

Mobile Food Truck Ordering System

Project Description:

A food truck is a large vehicle such as a van. Mobile catering is the business of selling food from some sort of vehicle. Many types of foods may be prepared. These days people have times when they are unable to cook. It will be used by people. Mobile catering is also used to provide the best food to people with less price. In this our project we have three modules.

- They are:-
1. Food Truck
 2. Delivery Boy
 3. Customer

Food truckers are registered into the website with their details like name, valid email, valid phone number etc... After completion of registration they login into the website with their valid email id and password. The food trucker can update timings and locations which place they are available on a day. The food trucker can add the food categories on the website. The food truckers can view the food categories and they can add the food items into the various categories. They can view the added food items. The Food truck can mark food items status as enable/disable. When a food truck disables the food item, it's not going to be visible for the customer.

Customers can register into the website with their valid details like name, valid email id, phone number etc... after completion of their registration they login into the website with their valid information like email, password. When they login into the website they browse trucks near by the locations. The customer can view truck menus. The Customers can add food items into the cart. They can order the food with their location details with payment. Payment method can be online which is either through card or paypal, Apple pay etc. and customers can view the orders. If customers want they can cancel the order before preparing the food.

The food truck can view the orders. The food truck makes order status as preparing means they are cooking the food. Then after cooking they will mark as food prepared. When food is prepared, the food truck sends a message to the delivery boys that the food has been

prepared. The food truck doesn't have any particular delivery boys. Many food trucks together are associated with a website which has individual delivery boys registered in it. The food delivery request will rise to the delivery boys. The delivery boy can register into the website with their own valid information like name, phone number, email etc..next delivery boy compilation of their registration they must and should login into the website with valid email and password. If the delivery boy accepts the food truck message first, they will take the order. The delivery boy received the request for food delivery from the food truck. The delivery boy can view the order request and accept the order. If one delivery boy accepts the food delivery request, then the request will be suspended for the other delivery boys. The delivery boys are to take the food items from the food truck to the customers in time and when the customers receive the order. The customer will mark the order as received. After order status is received, the customers can provide review and rating for the food items. This review and rating will help other customers to investigate before ordering the food.

Functionalities:

Food Truck

- Register
- Login
- Update timings and locations
- Add/view food categories
- add/view food items
- enable/disable food items
- View orders
- Make order status as preparing
- Make order status as prepared

Delivery Boy

- Register
- Login
- View order request
- accept/reject order request
- Make order status as picked up
- Make order status as delivered

Customer

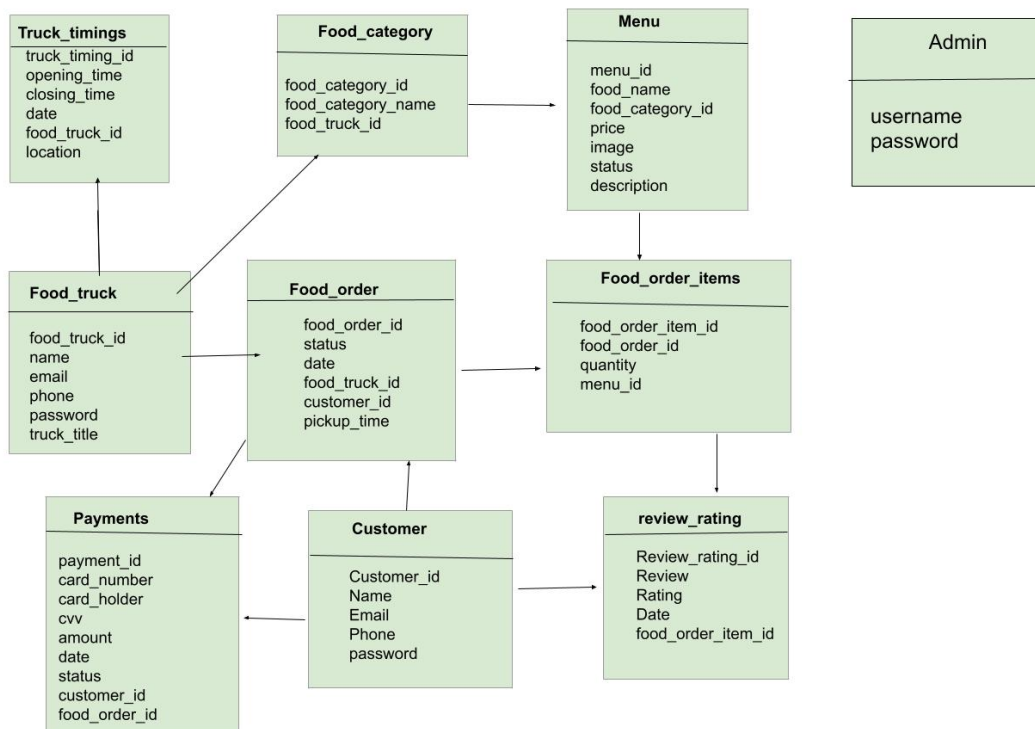
- Register
- Login
- Browse trucks with location wise
- View truck menu
- Add food items to cart
- Place order with payment
- View orders
- Cancel order
- Make status as received
- Give review rating for food items

DataBase Description:

In our project there are admin, truck_timings, food_truck, customer, food_category, Menu, food_order_items, food_order, review_rating, payment collections. The admin will login into the website with predefined login credential like username and password. The food_truck will register into the website with name, email, phone, truck_title. Then the details will be stored in the food_truck collection. The food_truck will add the location, opening time, closing time, date and food_truck_id is the reference key and these data will be stored in the truck_timings collection. The customer will register in to the website by name, email, phone these data will store in the customer collection. Then the customer can login into the website by using email and password. The food_truck will add the food_category by food_category_name, food_truck_id as reference key these data will store in the food category collection. Based on categories the food_truck will add the menu by food_name, price, image, description, status, food_category_id as reference key. These data will be stored in the menu collection. The customer can create the cart in that food_truck_id, customer_id, as reference keys, status, date, pickup_time here the customer can choose the pickup timings within the food truck timings. These data will be stored in the food_order collection. After that the customer can choose the food items, the quantity of the food, food_order_id, food_item_id as reference key; these data will be stored in the food_order_items collection. The customer, a food truck, can view the order items. The

customer will pay for the ordered food items by food_order_id as reference keys, date, amount, card_number, holder_name, status whether the payment is done or not. This data will be stored in the payment collection. The customer can give the review and rating for the food item by review, rating, date, food_item_id as reference key. These data will be stored in the review_rating collection.

DataBase Diagram:



Data Dictionary:

<div>customer</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 2</div><div>Avg. document size: 107.00 B</div><div>Indexes: 1</div><div>Total index size: 36.86 kB</div></div>	<div>food_category</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 81.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div>	<div>food_order_items</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 86.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div>	<div>food_orders</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 141.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div>
<div>food_truck</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 2</div><div>Avg. document size: 140.00 B</div><div>Indexes: 1</div><div>Total index size: 36.86 kB</div></div>	<div>menu</div> <div><div>Storage size: 4.10 kB</div><div>Documents: 1</div><div>Avg. document size: 22.00 B</div><div>Indexes: 1</div><div>Total index size: 4.10 kB</div></div>	<div>payment</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 201.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div>	<div>review_rating</div> <div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 131.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div>
<div>admin</div> <div><div>Storage size: 4.10 kB</div><div>Documents: 1</div><div>Avg. document size: 22.00 B</div><div>Indexes: 1</div><div>Total index size: 4.10 kB</div></div>	<div>truck_timings<div></div><div><div>Storage size: 20.48 kB</div><div>Documents: 1</div><div>Avg. document size: 150.00 B</div><div>Indexes: 1</div><div>Total index size: 20.48 kB</div></div></div>		

Sample Data:

Admin

```
▼ {
  ▼ {
    "_id": {},
    "$oid": "653744817849a88f70fb9519"
  },
  "password": "admin",
  "username": "admin"
}
```

Food Truck

```
▼ {
  ▶ {
    "_id": {},
    "Truck_title": "watson foods",
    "email": "watson@gmail.com",
    "name": "watson",
    "password": "xyz",
    "phone": "64587837894"
  }
}
```

Truck Timings

```
▼ {
  ▶ {
    "_id": {},
    "closing_time": "10:00 pm",
    "date": "25/09/2023",
    ▶ {
      "food_truck_id": {},
      "location": "vatican city",
      "opening_time": "9:00 am"
    }
  }
}
```

Customer

```
▼ {
  ▶ {
    "_id": {},
    "email": "john@gmail.com",
    "name": "john",
    "password": "xyz",
    "phone": "56739747873"
  }
}
```

Food Category

```
▼ {
  ▶ "_id": {...},
    "food_category_name": "non-veg",
  ▶ "food_truck_id": {...}
}
```

Menu

```
▼ {
  ▶ "_id": {...},
    "description": "it is a non veg item ",
  ▶ "food_category_id": {...},
    "food_item_name": "chicken manchooria",
    "image": "file.jpg",
    "price": "$20",
    "status": "available"
}
```

Food Orders

```
{
  "_id": {...},
  "customer_id": {
    "$oid": "6512ce2f1917c2ae485f1a50"
  },
  "date": "30/09/2023",
  "food_truck_id": {
    "$oid": "6512ceab1917c2ae485f1a52"
  },
  "status": "ordered",
  "pickup_time": "12:30 pm"
}
```

Food Order Items

```
{
  "_id": {},
  "food_order_id": {
    "$oid": "6512d1931917c2ae485f1a60"
  },
  "quantity": "2",
  "menu_id": {
    "$oid": "6512d0551917c2ae485f1a5c"
  }
}
```

Payments

```
▼ {
▼   "_id": {
      "$oid": "6512d3311917c2ae485f1a6f"
    },
▼   "customer_id": {
      "$oid": "6512ce2f1917c2ae485f1a50"
    },
    "date": "30/09/2023",
    "status": "paid",
    "Amount": "$20",
    "card_number": "6575837345654444",
    "cvv": "4564",
    "holder_name": "richard",
▼   "food_order_id": {
      "$oid": "6512d1931917c2ae485f1a60"
    }
}
```

Review Ratings


```
{
  "_id": {},
  "food_item_id": {
    "$oid": "6512d0551917c2ae485f1a5c"
  },
  "rating": "4",
  "review": "good",
  "date": "17/10/2023",
  "customer_id": {
    "$oid": "6512ce2f1917c2ae485f1a50"
  }
}
```

Userinterface and Forms

Home Page

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Welcome To Food Truck Management



This is the mobile food truck ordering system home page where you can see the admin, food truck and customer.

Admin Login page

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Admin Login

User Name

Password

Login

This is the admin page where the admin is going to enter username and password after entering the credentials by clicking on login the admin will enter into admin page.

Admin Home Page

[Home](#) [View Food Trucks](#) [View Customers](#) [Logout](#)

welcome Administrator

This is the homepage of the admin where you can see view food trucks and view customers in this page.

Food Truck Login

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Food Truck Login

Email

Enter Email Address

Password

Enter password

Login

New Food Truck ? [click here to register](#)

This is the food truck page where the food truck is going to enter email and password after entering the credentials by clicking on login the food truck will enter into the food truck page. If the food truck is not yet registered then by clicking click here to register is going to be redirect to the registration page.

Food Truck Registration

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

New Food Truck Registration

Food Truck Title

Enter Food Truck Title

Name

Enter Name

Email

Enter Email

Password

Enter password

Confirm Password

Enter confirm_password

phone

Enter Phone Number

Address

Enter address

Register

Already has account ? [Login Here](#)

For new registration of food trucks they have to enter the food truck title, name, email, password, confirm the password , phone number and address of the person who is going to be registered and then by clicking on register the truck will be registered. If the truck is already registered then click on the login here which will redirect to the login page.

Email verification for the food truck

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Email Verification

OTP(You receive to your email)

verify

For the new registration of food trucks the email verification is going to be done. By entering the otp sent to the email and then registration will be succeed.

Admin verification of the Food truck

[Home](#) [View Food Trucks](#) [View Customers](#) [Logout](#)

View Food Trucks					
Food Truck Name	Food Truck Title	Email	Phone	Verification Status	Verification
keerthi	Keethi Truck	keerthikeerthana89778@gmail.com	9485048483	OTP verified	<button>verify</button>

The admin is going to verify the new food trucks by checking off the details and then by clicking on verify the food truck will be verified. After verification the status will be updated to verified.

Food Truck Home page and Profile

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

welcome keerthi



Title:- Keethi Truck
Name:- keerthi
Email:- keerthikeerthana89778@gmail.com
Phone Number:- 9485048483
Verification Status:- Verified

This is the homepage of the food truck and profile of the food truck with details in the home page. We can view the truck timings, categories, food orders, dispatched and history.

Food truck timings

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

Add Food Truck Timings

From Time

--:--

To Time

--:--

Date

dd-mm-yyyy

Location

Enter Location Name

Add Timings

Timings Added Successfully

Truck Timings

05-12-2023

From Time	To time	Date	Location
09:00 AM	12:00 PM	2023-12-05	us

In this page the food truck is going to add the timings of the food truck. In adding timings of from time to end time, date and location of the food truck. On the same page the food truck can view the timings based on the date selection.

Add Category in food truck

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

Add New Category

Category Name

Enter Category Name

Add Category

Category added successfully

Categories List

Category Name
veg
non-veg

In this page the food truck is going to add the categories present in the food truck. On the same page food trucks can view the categories added to the food truck.

Add food items to the food truck

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

Add Food Items

Food Item Name

Enter Food Item Name

Price

Enter price

Quantity

Enter quantity

Units

Enter units

Category

Choose Category

Picture

Choose File

No file chosen

Description

Enter Description

Add Food Item

Search Here

search food name,category

chicken


\$ 100

2 plate

Available

non-veg

No Rating



This chicken biryani recipe features succulent, juicy pieces of chicken that are cooked in a yogurt marinade and then layered with crispy onions, coriander, mint and basmati rice to give you a dish that the entire family will enjoy.

idli


\$ 30

4 plates

Available

veg

No Rating



If the batter turns up hot, idli may turn hard. 3. Pour 2 to 4 tbsps more water if needed and blend till you get frothy thick smooth batter

In this page a food truck is going to add the food items and also in the same page added items can be viewed. While adding food items they have to enter the food item name, price, quantity, units, choose category, description and choose the picture of the food item. By choosing a food item or food category the food truck can view the items present in the food truck.

Customer Login

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Customer Login

Email

Enter Email Address

Password

Enter password

Login

New Customer ? [click here to register](#)

This is the customer page where the customer is going to enter email and password after entering the credentials by clicking on login the customer will enter into the customer page. If the customer is not yet registered then by clicking click here to register is going to be redirect to the registration page

Customer Registration

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

New Customer Registration

Name

Enter Name

Email

Enter Email

Password

Enter password

phone

Enter Phone Number

Gender ☐ Male ☐ Female

Address

Register

Already has account ? [Login Here](#)

For new registration of customers they have to enter the customer name, email, password, gender, phone number and address of the person who is going to be registered and then by clicking on register the customer will be registered. If the customer is already registered then click on the login here which will redirect to the login page.

Customer verification with email

[Home](#) [Admin](#) [FoodTruck](#) [Customer](#)

Email Verification

OTP(You receive to your email)

verify

For the new registration of customers the email verification is going to be done. By entering the otp sent to the email and then registration will be successful.

Customer Search Food


[Home](#) [Search Food](#) [Search Truck](#) [Cart](#) [Orders](#) [History](#) [Logout](#)

Search Here

search food name,category

chicken

\$ 100
2 plate
Available
non-veg
No Rating



-

1


+

+ADD

This chicken biryani recipe features succulent, juicy pieces of chicken that are cooked in a yogurt marinade and then layered with crispy onions, coriander, mint and basmati rice to give you a dish that the entire family will enjoy.

idli

\$ 30
4 plates
Available
veg
No Rating



-

1

+

+ADD


If the batter turns up hot, idli may turn hard. 3. Pour 2 to 4 tbsps more water if needed and blend till you get frothy thick smooth batter

In this page the customer can have a search option for food items which are available in the food trucks. By clicking on the add button then the items will be added to the cart.

Customers Search for trucks

[Home](#) [Search Food](#) [Search Truck](#) [Cart](#) [Orders](#) [History](#) [Logout](#)

idli
\$ 30
4 plates
Available
veg
No Rating



-

1

+

+ADD

If the batter turns up hot, idli may turn hard. 3. Pour 2 to 4 tbsps more water if needed and blend till you get frothy thick smooth batter

In this page the customer can search for the truck by choosing the truck name or category or food items.

Cart

[Home](#) [Search Food](#) [Search Truck](#) [Cart](#) [Orders](#) [History](#) [Logout](#)


Order Id
656eb8bc2e930f5e8b47af05

Customer
ravi, 123

Food Truck
Keethi Truck

Date And Time
2023-12-05 11:14:28.297000

Status
cart

Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price	Remove
	idli	veg	Keethi Truck	1	\$ 30	remove
					Total Amount:\$30	

Pickup Time
11:18

Choose Truck Location & Timings
2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us

Order Now

In this page the customer can view the items added to the cart. If the customer is not willing to order then the customer can remove the items from the cart. At last the customer can order the items by clicking on order now.

Payment

[Home](#) [Search Food](#) [Search Truck](#) [Cart](#) [Orders](#) [History](#) [Logout](#)

Payment Details

Card Type

Choose Card Type

Card Number

Enter Card Number

Card Holder Name

Enter Card Holder Name

CVV

Enter CVV Number

Expiry Date

Enter Expiry Date

Amount

\$30

pay amount

In this page the customer can do a payment option for the order items. By entering the card type, card number, card holder name, cvv, expiry date and amount have to be paid. By clicking on the pay amount the payment will be done.

Orders

[Home](#) [Search Food](#) [Search Truck](#) [Cart](#) [Orders](#) [History](#) [Logout](#)

Order Id
656eb8bc2e930f5e8b47af05


Customer
ravi, 123

Food Truck
Keethi Truck

Date And Time
2023-12-05 11:14:28.297000

pickup Time
11:18

Status
ordered

Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price
	idli	veg	Keethi Truck	1	\$ 30
					Total Amount:\$30

Truck Timings
2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us

Cancel

In this page the customer can view the orders of the customer. If the customer is not willing to take the order then the customer can cancel the order.

Food Truck orders

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

Order Id
656eb8bc2e930f5e8b47af05


Customer
ravi, 123

Food Truck
Keethi Truck

Date And Time
2023-12-05 11:14:28.297000

pickup Time
11:18

Status
ordered

Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price
	idli	veg	Keethi Truck	1	\$ 30
					Total Amount:\$30

Truck Timings
2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us

Make As Cook

In this page the food truck can view the orders received from the customers. If the food is ready for the customer then the food truck will update the status of the order by clicking on make a cook. It will be updated to cook.

Dispatch of order

[Home](#) [Truck Timings](#) [Categories](#) [Food Items](#) [Orders](#) [Dispatched](#) [History](#) [Logout](#)

Order Id
656eb8bc2e930f5e8b47af05


Customer
ravi, 123

Food Truck
Keethi Truck

Date And Time
2023-12-05 11:14:28.297000

pickup Time
11:18

Status
Ready for pickup


Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price
	idli	veg	Keethi Truck	1	\$ 30
					Total Amount:\$30

Truck Timings
2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us

Dispatch

In this page the food truck will dispatch the order by clicking on the dispatch.


Customer orders

Order Id 656eb8bc2e930f5e8b47af05	Customer ravi, 123	Food Truck Keethi Truck	Date And Time 2023-12-05 11:14:28.297000	pickup Time 11:18	Status Dispatched
Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price
	idli	veg	Keethi Truck	1	\$ 30
					Total Amount:\$30
Truck Timings 2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us					
Make As Receive					

After receiving the order the customer will update the status of the order by clicking on the make as receive .


Customer history

Home Search Food Search Truck Cart Orders History Logout

Order Id 656eb8bc2e930f5e8b47af05	Customer ravi, 123	Food Truck Keethi Truck	Date And Time 2023-12-05 11:14:28.297000	pickup Time 11:18	Status delivered	
Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price	Review
	idli	veg	Keethi Truck	1	\$ 30	Review
					Total Amount:\$30	
Truck Timings 2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us						

In this the customer can view the history of the order placed in the past. With full details of the order.

Food truck History

Home Truck Timings Categories Food Items Orders Dispatched History Logout					
Order Id 656eb9fb2e930f5e8b47af07	Customer ravi, 123	Food Truck Keethi Truck	Date And Time 2023-12-05 11:19:47.670000	pickup Time 11:23	Status Dispatched
Food Item Image	Food Item Name	Category Name	Food Truck Name	Quantity	Price
	chicken	non-veg	Keethi Truck	1	\$ 100
					Total Amount:\$100
Truck Timings 2023-12-05 09:00:00 - 2023-12-05 12:00:00 available at us					

In this page the food truck can view the history of the orders placed by the customers. With full details of the order with customer name, ordered item, quantity and price.

Source Code:

```
1  import datetime
2
3  import os
4  import re
5
6  from flask import Flask, request, render_template, session, redirect
7  from bson import ObjectId
8  import pymongo
9  my_collections = pymongo.MongoClient("mongodb://localhost:27017/")
10 my_db = my_collections['MOBILE_FOOD_TRUCK_MANAGEMENT_SYSTEM']
11 admin_col = my_db['Admin']
12 food_truck_col = my_db['Food_Truck']
13 customer_col = my_db['Customer']
14 truck_timings_col = my_db['Truck_Timings']
15 category_col = my_db['Categories']
16 food_item_col = my_db['Food_Items']
17 customer_order_col = my_db['Customer_Orders']
18 customer_order_item_col = my_db['Customer_Order_Items']
19 review_col = my_db['Review']
20
```

This is the main.py page connection with mongodb, and import flask, import pymongo, import bson packages all install the packages, And collections are created in python.

Python method:

```
@app.route("/")
def home():
    return render_template("home.html")

@app.route("/admin_login")
def admin_login():
    return render_template("admin_login.html")
```

It's creates an index method and returns a index.html page to the python page.

Html page:

```

1 {% include "head.html"%}
2
3 <div class="row">
4   <div class="col-md-4 p-3 "></div>
5   <div class="col-md-4 p-3 ">
6     <div class="card p-3">
7       <form action="admin_login1" method="post" id="my-details">
8         <h2 class="text-center">Admin Login</h2>
9         <div class="form-group mt-3">
10          <label>User Name</label>
11          <input type="text" name="username" id="username" class="form-control" placeholder="Enter User Name">
12          <label style="color:red"></label>
13        </div>
14        <div class="form-group mt-3">
15          <label>Password</label>
16          <input type="password" name="password" id="password" class="form-control" placeholder="Enter password">
17          <label id="password-msg" style="color:red"></label>
18        </div>
19        <div class="mt-3">
20          <input type="submit" value="Login" class="btn-home">
21        </div>
22      </form>
23    </div>
24  </div>
25 </div class="col-md-4 p-3 "></div>
26 </div>

```

This is the html page the admin will add the username and password and login button on the website.

```

@app.route("/admin_login")
def admin_login():
    return render_template("admin_login.html")

@app.route(rule="/admin_login1", methods=['post'])
def admin_login1():
    username = request.form.get("username")
    password = request.form.get("password")
    query = {"username": username, "password": password}
    admin = admin_col.find_one(query)
    if admin != None:
        session['admin_id'] = str(admin['_id'])
        session['role'] = 'Admin'
        return redirect("/admin_home")
    else:
        return render_template(template_name_or_list="msg.html", message="Invalid Login Details", color="text-danger")

@app.route("/admin_home")
def admin_home():
    return render_template("admin_home.html")

```

Def admin login method is the values are get and return response in the message page add on the website

```

@app.route("/food_truck_registration")
def food_truck_registration():
    return render_template("food_truck_registration.html")

@app.route(rule="/food_truck_registration_action", methods=['post'])
def food_truck_registration_action():
    food_truck_title = request.form.get("food_truck_title")
    name = request.form.get("name")
    email = request.form.get("email")
    password = request.form.get("password")
    confirm_password = request.form.get("confirm_password")
    phone = request.form.get("phone")
    address = request.form.get("address")
    query = {"email": email}
    count = food_truck_col.count_documents(query)
    if count > 0:
        return render_template(template_name_or_list="msg.html", message="duplicate email address")
    query = {"phone": phone}
    count = food_truck_col.count_documents(query)
    if count > 0:
        return render_template(template_name_or_list="msg.html", message="duplicate phone number")
    try:
        query = {"food_truck_title": food_truck_title, "name": name, "email": email, "password": password, "confirm_password": confirm_password, "phone":
        result = food_truck_col.insert_one(query)

```

```

        otp = random.randrange(start=1000, stop=100000)
        subject = "Food truck email verification"
        message = "Use this OTP "+str(otp)+" to verify your account"
        print(otp)
        send_email(subject, message, email)
        return render_template(template_name_or_list="food_truck_verification.html", food_truck_id=food_truck_id, otp=otp)
    except Exception as e:
        print(e)
        return render_template(template_name_or_list="msg.html", message="something went wrong")

@app.route(rule="/food_truck_verification", methods=['post'])
def food_truck_verification():
    food_truck_id = request.form.get("food_truck_id")
    query = {"_id": ObjectId(food_truck_id)}
    query2 = {"$set": {"verification_status": 'OTP verified'}}
    food_truck_col.update_one(query, query2)
    return render_template(template_name_or_list="msg.html", message="food registered successfully")

@app.route("/verify_food_truck")
def verify_food_truck():
    food_truck_id = request.args.get("food_truck_id")
    query = {"_id": ObjectId(food_truck_id)}
    query2 = {"$set": {"verification_status": 'Verified'}}
    food_truck_col.update_one(query, query2)
    return render_template(template_name_or_list="msg.html", message="food verified successfully")

```