



CMPE 280 Web UI Design

# Foodstrap

MAKING IT EASY TO DONATE FOOD

Submitted to  
Prof. Chandrasekar Vuppalapati

Submitted by  
Mayura Dhivya Nehruji (Id: 012434461)  
Parvathy Kannankumarath Madom Krishnan (Id: 014533623)

# Contents

1. Project Description.....	4
2. Application Set Up Instructions .....	5
3. Team Member Contribution .....	6
4. Requirements .....	6
01. Functional Requirements .....	7
02. Non-functional requirements .....	8
5. Web UI Requirement Principles .....	9
01. Personas .....	9
02. Job Shadowing and Contextual Interviews.....	11
03. Activity Centered Design.....	13
04. Text Usability .....	13
05. Mental Models .....	14
6. Web UI Design Principles .....	14
01. Flow Diagram.....	14
02. Storyboards .....	15
001. Restaurant Manager.....	15
002. Shelter Manager.....	17
003. Volunteer.....	20
03. Sketches.....	22
04. Wireframes .....	29
05. Application Features .....	37
7. High Level Architecture Design .....	38
01. Technology Stack .....	38
8. Data Flow Diagram .....	39
01. Restaurant Manager .....	40
02. Shelter Manager .....	41
03. Volunteer .....	42
9. Component Diagram.....	43
10. Workflows.....	43
01. Restaurant .....	43
02. Shelter .....	44

03.	Volunteer .....	44
11.	Sequence Diagram.....	45
12.	HTML5 Features .....	46
13.	Server-Side Design .....	53
01.	APIs.....	53
02.	DB Design .....	56
14.	Client-Side Design.....	58
15.	Test Automation .....	61
01.	Test Automation using Mocha.....	61
02.	Selenium Automation.....	62
001.	Set up.....	62
002.	Automation Scripts .....	63
16.	Load Testing using JMeter .....	70
17.	Cross Browser Compatibility.....	74
01.	Cross Device Compatibility .....	76
18.	Library .....	79
01.	Front end JavaScript Library .....	79
02.	NodeJS Library .....	85
03.	Library used in Test Automation.....	89
19.	Design Patterns.....	91
20.	Pagination .....	93
21.	Search Engine Optimization .....	95
22.	Profiling .....	98
01.	Heap snapshot.....	98
02.	Performance Profiling .....	101
23.	Localization .....	102
01.	Text Localization .....	104
02.	Image Localization .....	107
24.	Screenshots .....	109
01.	Application .....	109
02.	Localization.....	123
001.	Hindi.....	123
002.	French.....	132

# 1. Project Description

America wastes roughly 40 percent of its food. Of the estimated 125 to 160 billion pounds of food that goes to waste every year, much of it is perfectly edible and nutritious. Uneaten food puts unneeded strain on the environment by wasting valuable resources like water and farmland. At a time when 12 percent of American households are food insecure, reducing food waste by just 15 percent could provide enough sustenance to feed more than 25 million people, annually. Currently, only 10 percent of edible wasted food is recovered each year, in the US. Barriers to recovering food are liability concerns, distribution and storage logistics, and funds needed for gleaning, collecting, packaging, and distribution.

An enormous amount of food is wasted or thrown out at restaurants across the country every single day. US restaurants generate an estimated 22 to 33 billion pounds of food waste each year. Kitchen culture and staff behavior such as over-preparation of food, improper ingredient storage and failure to use food scraps and trimmings contribute to food loss. All-you-can-eat buffets are particularly wasteful. The first step toward reducing the amount of food wasted is to change the behavior of the public, as well as chefs and workers at these restaurants, by finding better and smarter ways to use this food in the kitchen and diverting the food to the homeless shelters.

FoodStrap, a web application, aims to bridge the gap between hunger and food waste. It makes it easy to donate leftover and surplus food. The application provides a platform where all the stakeholders for food donation can participate and make the process easy. Through the application, restaurants can donate leftover food at the end of each day and connect with homeless shelters and volunteers in its community. Homeless shelters can sign up in the application and claim food donated by restaurants. Philanthropic people can register themselves as volunteers for homeless shelters and can drive for them. They can pick up the food from restaurants and deliver them to the shelters.

A restaurant manager can sign up for his restaurant in the app. At the end of each day or whenever a food audit is done, the manager can assess the quality and quantity of left-over food. If the food is edible and of a good grade, he can donate this to homeless shelters in his locality. He can weigh, package and log the food. All the food items should be individually packed and clearly labeled for ingredients and allergy information. Then he can submit a donation in the app specifying all the details like menu, portion count, pick up time, and address. Allergy information and any special notes for pick up can also be mentioned. He can then track the status of the donation through history in the app. Later a volunteer picks up the food from the restaurant. The manager can see all the donations from the past. He can update the profile of the restaurant and also see the detailed steps for donating. To provide a good user experience, charts are present in the dashboard and history views to show the number of people fed by the restaurant through the donations.

Homeless shelters that provide food assistance to people in need, can sign up in the application. The shelter manager can search in the application for food donations in the locality. If any donation matches

the manager's criteria for menu and portion count, he can then claim the donation. He updates the status of the donation in the app to claimed. The manager should consider the allergy information and pick notes if any. Later the food is delivered to the shelter from the restaurant by volunteers. The manager can check maps and charts to get an idea about the number of donations received, the donor restaurant location, etc. He can update his profile information also.

Altruistic people can sign up in the application to be volunteers for a homeless shelter. They will drive for the homeless shelters picking up food donations from restaurants and delivering it to shelters. A volunteer willing to drive, can sign into his dashboard and check for any donations claimed by his shelter. He can then update the status of the donation to pick up if he can pick up and deliver that donation. The volunteer can see all the required information for pick up from the donation form like pick up time and address and notes if any. He can see the map showing the pickup address of the donor restaurant.

The application is developed using NodeJS. For the front end, EJS Embedded JavaScript templates with HTML5 and CSS3 are used. Libraries like jQuery, Bootstrap, Google Maps, and Google Charts are also used to enhance the UI. APIs are developed using Express.js and MongoDB is the database used. It is hosted on AWS EC2 instances. Test case automation for APIs is done using Mocha. Selenium automation is done for the overall flow. Load and performance testing is done using JMeter. Profiling is done using Chrome tools like CPU profiling and heap snapshot. Cross-browser compatibility is also verified. The application was designed using MVC pattern with each layer separated from one another. Search engine optimization techniques are implemented to make it easy to search for the application on the web. The application is developed with localization support for English, Hindi, and French locales.

## 2. Application Set Up Instructions

### Pre-requisites

- A machine with NodeJS installed
- MongoDB installed and service running

### Set Up Instructions

- Unzip the source code zip folder and go to FOODSTRAP folder
- cd FOODSTRAP/foodstrap
- Install dependencies using npm
  - npm install
- Start the service
  - npm start

Service running in <http://<hostname>:5000/>

#### Run Mocha Test Cases

- npm test

#### Run Selenium test cases

- Detailed instructions are provided in Selenium Automation section.

## 3. Team Member Contribution

Task	Team Member
Research	Mayura, Parvathy
Design	Mayura, Parvathy
Landing Pages	Mayura, Parvathy
Restaurant manager Flow	Parvathy
Shelter Flow	Mayura
Volunteer Flow	Parvathy
Mocha Automation	Parvathy
Selenium Automation	Mayura
JMeter Load Testing	Parvathy
Profiling	Parvathy
Localization	Mayura, Parvathy
Cross Browser Compatibility	Mayura
Project Report	Mayura, Parvathy

## 4. Requirements

The first step in the development of any application is requirement gathering. Requirement gathering falls under the research part of the Web UI design and development. A requirement is a service or function to be provided by the application or system. The requirements can be categorized as functional and non-functional requirements. Functional requirements describe what the application must do or perform. Nonfunctional requirements are more like quality attributes or properties of the application.

The functional requirements acquired through applying the requirement principles as explained in the section titled Web UI Requirement Principles are listed below:

## 01. Functional Requirements

The functional requirements are listed in the table below

Number	Description	Priority
<b>1.APPLICATION</b>		
1.1	Application must allow new users (restaurant manager, shelter manager or volunteer) to sign up for use of the application	High
1.2	Application must allow existing users with valid credentials to log in to the application.	High
1.3	Application must allow the restaurant manager to donate food and to specify details such as menu, pickup address or time and special instructions	High
1.4	Application must allow the restaurant manager to track the status of the donation	High
1.5	Application must allow the restaurant manager to update the restaurant's details such as email	Medium
1.6	Shelter manager should be able to view the list of all open donations	High
1.7	Application should enable the shelter manager to claim an open donation one at a time	High
1.8	Application should list the status of claimed donations as well as those that are picked up, for status tracking	High
1.9	Application should facilitate the updating of shelter profile details such as email address	Medium
1.10	Application should let the volunteers associate with a shelter	High
1.11	Application must show a list of donations claimed by the shelter with which the volunteer is	High

	associated	
1.12	Volunteer should be able to pick up a claimed donation	High
1.13	Application should allow the volunteer profile details to be updated	Medium

## 02. Non-functional requirements

The non-functional requirements are listed in the table below

Number	Description	Priority
<b>1.Interface</b>		
1.1	The UI must be presented in English	High
1.2	The UI must be presented in other languages like Hindi	Medium
1.3	UI must follow a consistent theme across all pages	High
<b>2.Usability</b>		
2.1	More images, graphics, charts instead of long verbose texts	High
2.2	The UI must be easy to understand, making sure it's easy for the users to do what they want with little to no learning curve	High
2.3	The UI must have proper link text, so the users are able to navigate through the pages without any hassle.	High
<b>3.Perfomance</b>		
3.1	The application must support concurrent users without losing performance or crashing	High

## 5. Web UI Requirement Principles

### 01. Personas

#### **Jim Sanders, Sam 's Pizzeria Restaurant Manager**

Jim Halls is the manager in Sam's Pizzeria restaurant. In Sam's Pizzeria, several pounds of food are leftover at the end of each day. Jim decides to donate the leftover food to homeless shelters. By using FoodStrap, he connects with shelters and volunteers easily.

 <p><b>Name</b> Jim Sanders</p>	<p><b>Job Responsibilities</b></p> <ul style="list-style-type: none"> <li>• Managing restaurant</li> <li>• People management</li> <li>• Dispose off leftover food</li> <li>• Ensure leftover food is not wasted</li> </ul>	<p><b>They Gain Information By</b></p> <p>At the end of each day, the chef informs the manager about left over food</p>
<p><b>Age</b> 25 to 34 years</p>	<p><b>Reports to</b></p> <p>Head Chef Restaurant Owner</p>	<p><b>Goals or Objectives</b></p> <ul style="list-style-type: none"> <li>• Help customers with reservations and seating</li> <li>• Ensure wait staff cares for all customers</li> <li>• Maintain restaurant ambiance</li> <li>• Take care of left over food</li> </ul>
<p><b>Highest Level of Education</b> Bachelor's degree (e.g. BA, B!</p>		
<p><b>Industry</b> Food &amp; Beverage</p>	<p><b>Preferred Method of Communication</b></p> <ul style="list-style-type: none"> <li>• Phone</li> <li>• Text Messaging</li> <li>• Face-To-face</li> </ul>	<p><b>Their Job Is Measured By</b></p> <ul style="list-style-type: none"> <li>• Treating people with warmth and care</li> <li>• Ensuring left over food is not wasted</li> </ul>
<p><b>Organization Size</b> 1-10 employees</p>		

#### **Sally Sands, Manager, Home for Good Homeless Shelter**

Home for Good Homeless Shelter provides assistance to hundreds of homeless people. Sally, the manager, connects with local restaurants and grocery stores to obtain food donations to feed the people in the shelter. She uses the FoodStrap app to connect with local restaurants and obtains their surplus food.

 <p><b>Name</b> Sally Sands</p>	<p><b>Job Responsibilities</b></p> <ul style="list-style-type: none"> <li>Provide assistance to people in need</li> <li>Connect with local restaurants, grocery stores and other donors</li> <li>Collect food, grocery and donations</li> <li>Distribute items to people in need</li> </ul>	<p><b>They Gain Information By</b></p> <p>At the end of each day, restaurants inform Sally about left over food</p>
<p><b>Age</b> 30 to 40 years</p> <p><b>Highest Level of Education</b> Bachelor's degree</p> <p><b>Industry</b> Hospitality</p> <p><b>Organization Size</b> 1-10 employees</p>	<p><b>Their Job Is Measured By</b></p> <ul style="list-style-type: none"> <li>Treating people with warmth and care</li> <li>Ensuring left over food is claimed and served to the homeless people</li> </ul> <p><b>Preferred Method of Communication</b></p> <ul style="list-style-type: none"> <li>Phone</li> <li>Text Messaging</li> <li>Face-To-face</li> </ul>	<p><b>Goals or Objectives</b></p> <ul style="list-style-type: none"> <li>Help homeless people</li> <li>Ensure homeless people are assisted with food and shelter</li> <li>Collect donations in the form of food, grocery, clothes</li> <li>Maintain shelter atmosphere and hygiene</li> <li>Distribute collected donations</li> <li>Log all day to day details</li> </ul>
		<p><b>Reports to</b> Shelter Trusty</p>

### Patrick Jane, Volunteer at Home for Good

Patrick is a volunteer at Home for Good shelter. He spends quality time at the shelter helping the inmates with their tasks. He drives around town and collects donations for the shelter. He uses the FoodStrap app to keep track of all the donations. He typically spends few hours every week volunteering.

 <p><b>Name</b> Patrick Jane</p>	<p><b>Job Responsibilities</b></p> <ul style="list-style-type: none"> <li>Provide assistance to people in need</li> <li>Pick up donations from restaurants, caterers and grocery stores</li> <li>Deliver donations to shelters</li> <li>Distribute and serve homeless people</li> </ul>	<p><b>They Gain Information By</b></p> <p>At the end of each day, shelter informs Patrick about donations to be collected</p>
<p><b>Age</b> 20 to 30 years</p>	<p><b>Their Job Is Measured By</b></p> <ul style="list-style-type: none"> <li>Treating people with warmth and care</li> <li>Ensuring left over food is picked up and served to the homeless people</li> </ul>	<p><b>Goals or Objectives</b></p> <ul style="list-style-type: none"> <li>Help homeless people</li> <li>Ensure homeless people are assisted with food and shelter</li> <li>Pick up donations in the form of food, grocery, clothes</li> <li>Deliver donations at shelter</li> <li>Serve collected donations</li> </ul>
<p><b>Highest Level of Education</b> Bachelor's degree</p>		
<p><b>Industry</b> Hospitality</p>	<p><b>Preferred Method of Communication</b></p> <ul style="list-style-type: none"> <li>Phone</li> <li>Text Messaging</li> <li>Face-To-face</li> </ul>	<p><b>Reports to</b></p> <p>Shelter Manager</p>
<p><b>Organization Size</b> 10-50 employees</p>		

## 02. Job Shadowing and Contextual Interviews

A restaurant manager is shadowed and information about his job with respect to food waste is analyzed. Questions are asked to know about the difficulties in food waste and left-over food management. The following questions are asked to the manager

- On average, how much food is left over at the end of each day?
- On which all days of the week, more food is left over?
- On which all days of the month, more food is left over?
- How is left over food handled?
- Is left over food donated or thrown away?
- Are you aware of the food donation liabilities?
- Because of the food donation liabilities, are you hesitant to donate leftover food?
- Have you been sued because of food donation liabilities?
- How is left over food donated?
- Do you know shelters which accept leftover food?
- How do you find shelters to donate food?
- Do shelters come and check with the restaurant at the end of each day?
- Who delivers the leftover food?
- Or who picks up the leftover food?
- Have you ever visited the homeless shelter you are donating food to?

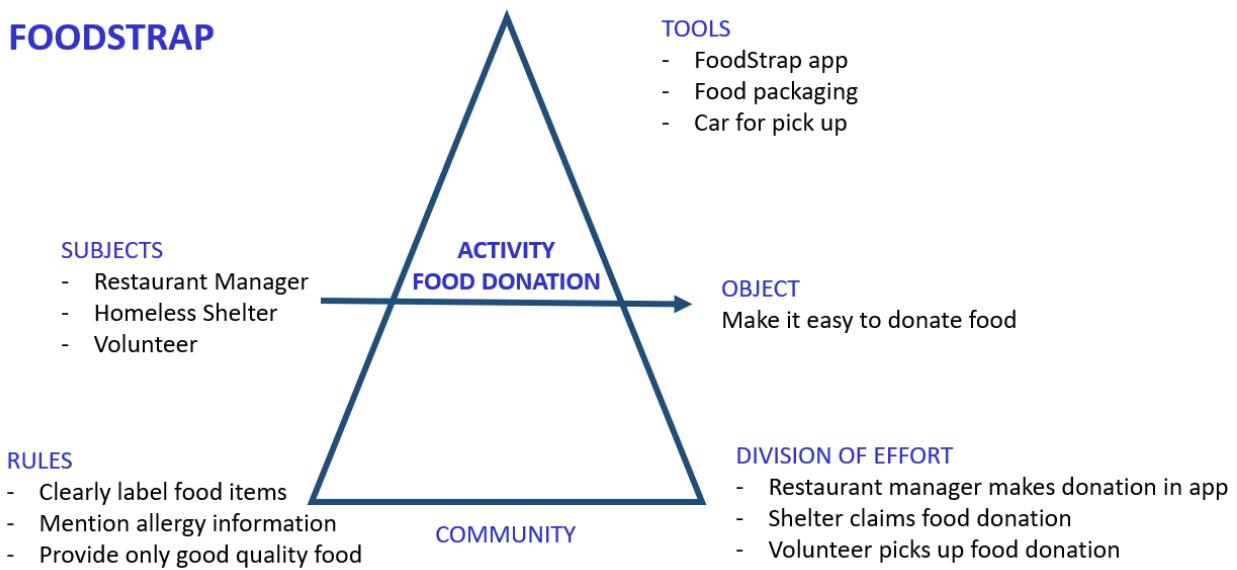
- How easy or difficult it is to donate leftover food?
- How easy or difficult it is to deliver leftover food to shelters?
- Do other uncooked food items become surplus?
- How do you donate such items?
- For food ordered by customers, do they finish their portion of food on the plate?
- For food ordered by customers, do they leave food on the plate?
- Can you reduce portion size to eliminate this food waste?

A homeless shelter manager is shadowed and information about his job with respect to collecting food donation is analyzed. Questions are asked to know about the difficulties in food collection and food donation management. The following questions are asked to the manager

- On average, how many people come to the shelter each day?
- On which all days of the week, more people come?
- On which all days of the month, more people come?
- Do you get the required donations to manage each day's work?
- From where do you get donations?
- What all items do you get and accept as donations?
- How many portions of food do you need to serve the people each day?
- Do you receive sufficient food donations each day?
- Do you reach out to restaurants regarding leftover food?
- Or do restaurants reach out to you?
- How is leftover food delivered?
- Do you go and collect them or do restaurants deliver?
- At what time of the day do you receive food donations?
- At what time do you need food to serve people?
- Are there enough volunteers to pick up and serve people?
- Do you log all donations made?
- Do you log all people visiting the shelter?
- How easy or difficult it is to get food donations?
- How easy or difficult it is to pick up food donations?
- How easy or difficult it is to serve and distribute food donations?
- Are you aware of food donation liabilities of restaurants?
- Does that hinder restaurants from donating food?
- Are food donations of good quality?
- Has there been any case of food allergies caused by food donated?
- Are food donations clearly labelled for its ingredients?
- Are food donations clearly labelled for any allergy information?

### 03. Activity Centered Design

The application is designed, and tailor made for the food donation activity. The focal point is the activity with users and other requirements around it. The main activity supported is food donation. The objective of the activity is to make food donation easy. The activity from different users' perspectives are also considered. The subjects are restaurant manager, homeless shelter and volunteer. The efforts for the activity are divided among the subjects. Restaurant manager makes donation through the app. Shelter claims food donations made by restaurants. Volunteer picks up claimed donations.



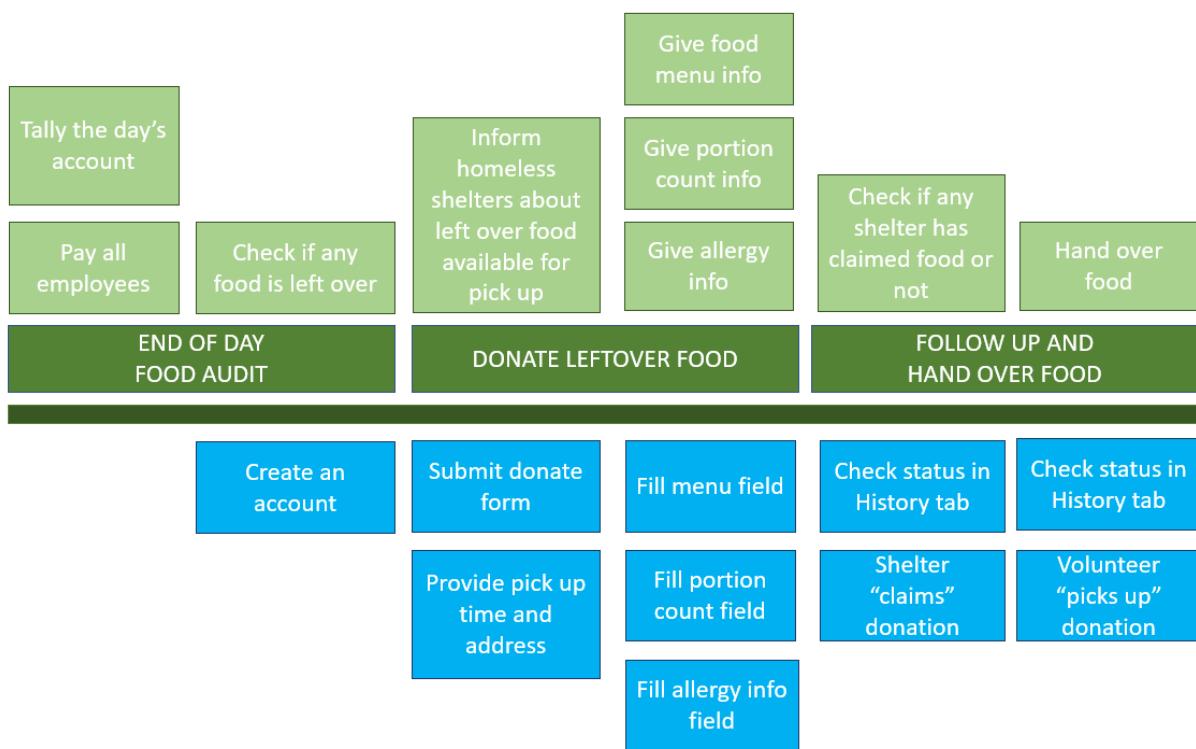
### 04. Text Usability

To provide good user experience and improve text usability, more images and graphs, and less text is used in the application. Images, graphs and maps convey ideas faster than text. For the limited amount of text present, certain guidelines were followed. The following guidelines are employed in the application development:

- Less text, more images and charts
- Sentences included are short and easy to read and understand.
- Headings are clear and easy to understand and demonstrate the content under it.
- The font sizes are sufficiently large, and fonts are readable.
- Number of fonts used is minimum
- Standard fonts are used
- Use of different colors for fonts is limited. All headings and text are either black or green
- Similar colors for text and background not used
- All pages follow same theme of green, white and black

## 05. Mental Models

A mental model of a user is what the user believes about the application. Individual users each have their own mental model. Mental models of developers may not be the same as those of their users. A mental model of a restaurant manager user is shown below.

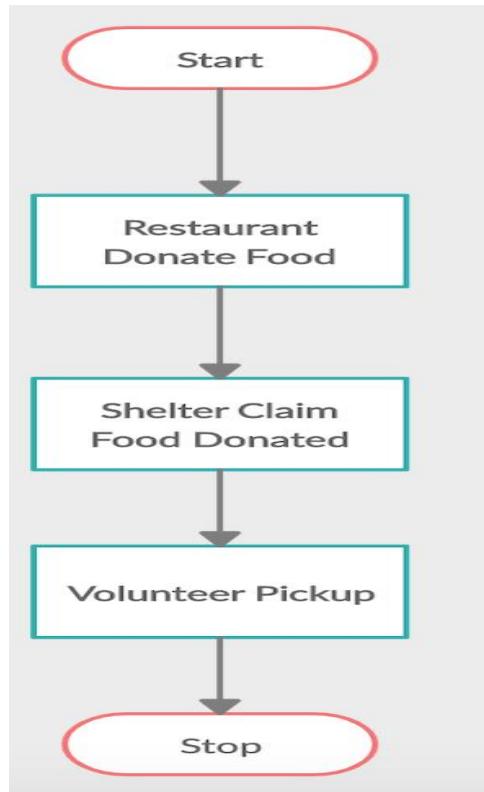


## 6. Web UI Design Principles

Web design principles or strategies used in the project are as follows

### 01. Flow Diagram

The first strategy is flow diagram. The flow diagram provides the workflow of the system. The high-level workflow is illustrated in the below figure.



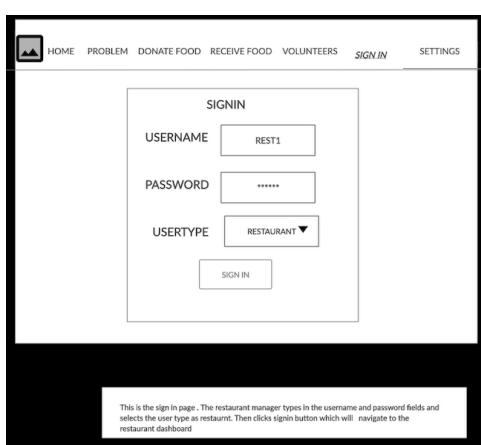
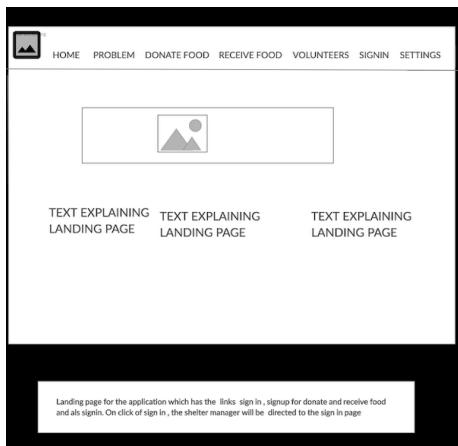
The detailed workflows for the processes of donating food, shelter claim and volunteer pickup are explained under the section titled Sequence and Workflow. The data flow diagrams are illustrated under the section titled Data Flow Diagram and Component Level Design.

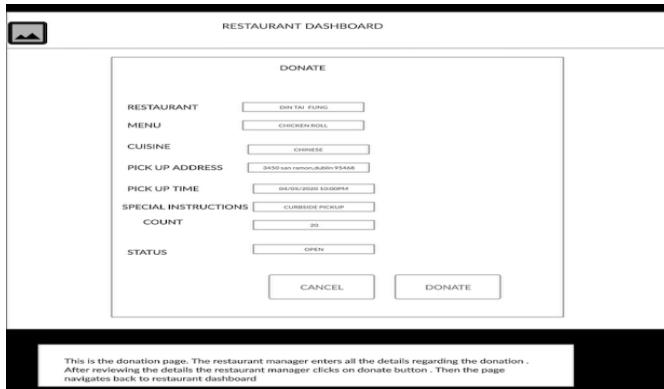
## 02. Storyboards

Storyboards for the main processes i.e. restaurant manager donating food, shelter manager claiming the donation and volunteer pickup of the donation can be found below.

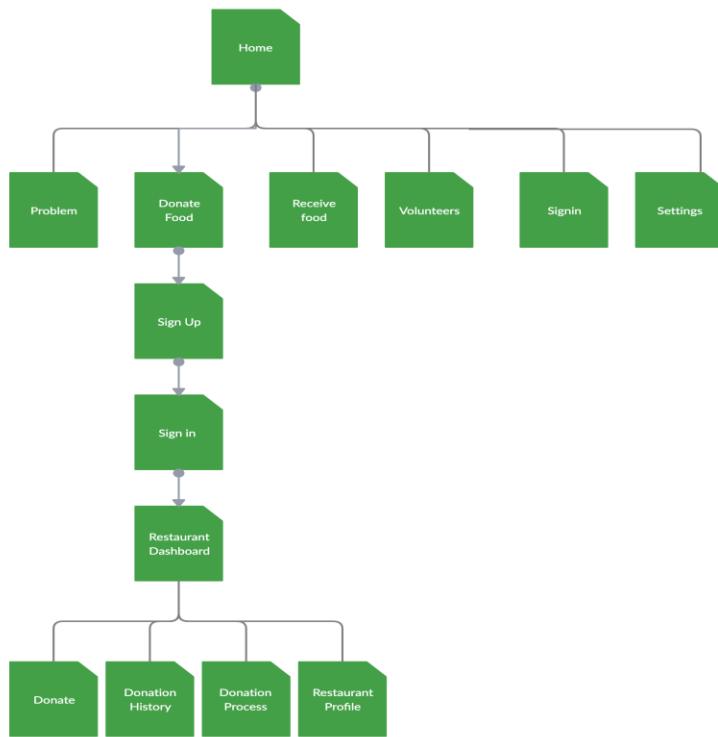
### 001. Restaurant Manager

STORYBOARD	PERSONA: RESTAURANT MANAGER	SCENARIO: RESTAURANT MANAGER MAKING FOOD DONATION
------------	--------------------------------	---





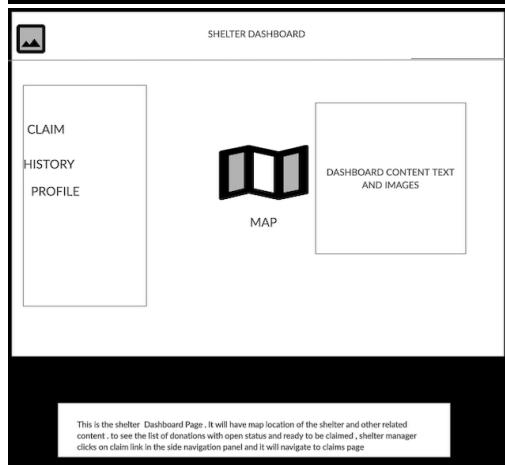
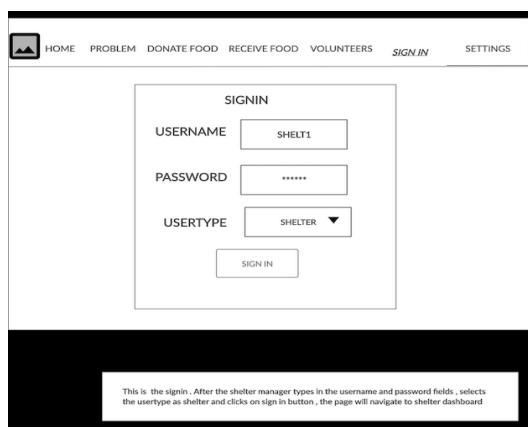
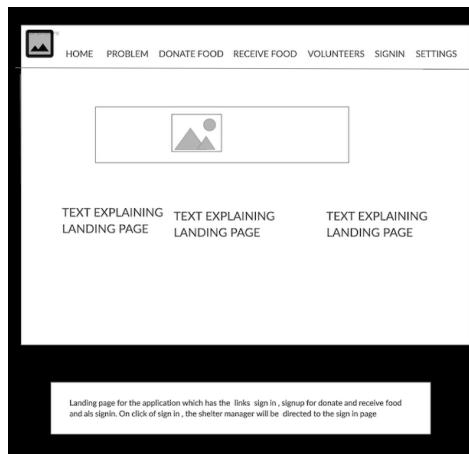
## GUI FLOW FOR RESTAURANT MANAGER



## 002. Shelter Manager

STORYBOARD

PERSONA: SHELTER  
MANAGERSCENARIO:  
SHELTER MANAGER CLAIMING  
DONATION



**SHELTER DASHBOARD**

**CLAIM HISTORY PROFILE**

RESTAURANT NAME	MENU	CUISINE	PICK UP ADDRESS AND TIME	SPECIAL INSTRUCTIONS	ALLERGY	STATUS	CLAIM
Din Tai Fung	Chicken roll	Chinese	3450 san ramon dublin ca 94568 04/05/2020 10:00 PM	curbside pick up	contains nuts	OPEN	<input type="button" value="CLAIM"/>

This is the claims page that will display a list of all the open food donations that is available for claim by the shelter. To claim a food donation , the shelter manager clicks on the CLAIM button corresponding to that donation item. The onclick navigates to the claim confirmation page.

**SHELTER DASHBOARD**

**CLAIM**

RESTAURANT

MENU

CUISINE

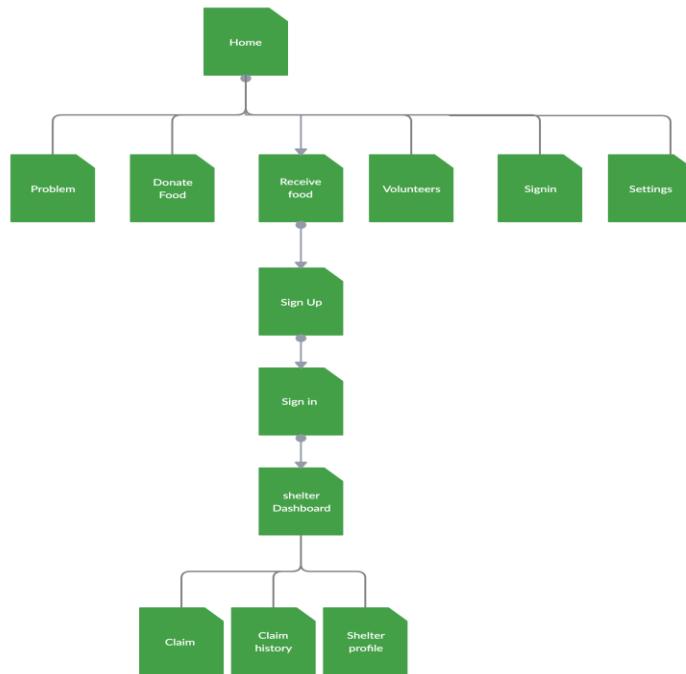
PICK UP ADDRESS

PICK UP TIME

SPECIAL INSTRUCTIONS

STATUS

This is the confirm claim page . It will display the details of the donation that the shelter manager wants to claim . After reviewing the details the shelter manager clicks on confirm claim button . Then the page navigates back to shelter dashboard

**GUI FLOW FOR SHELTER MANAGER:****003. Volunteer**

STORYBOARD	PERSONA: VOLUNTEER	SCENARIO: VOLUNTEER PICKING UP FOOD
		<b>SCENARIO: VOLUNTEER PICKING UP FOOD</b>
		Landing page for the application which has the links sign in , signup for donate and receive food and also volunteer signin. On click of sign in , the volunteer will be directed to the sign in page

This is the sign in page. After volunteer enters the username and password and select user type as volunteer and click on sign in button , the page will navigate to volunteers dashboard page

VOLUNTEER DASHBOARD

Name	Menu	Address	Pickup time	Sal instructions	Shelter	Status	Allergy	
Din tai fung	Chicken roll	3450 San Ramon Rd Dublin 94568	04/05/2020 10:00PM	curbside pickup	goodwill ambassador	CLAIMED	contains nuts	PICKUP

This is the volunteer Dashboard Page . It will have a list of donations claimed by the shelter the volunteer is associated with . On click of the pickup button , pickup page loads.

VOLUNTEER DASHBOARD

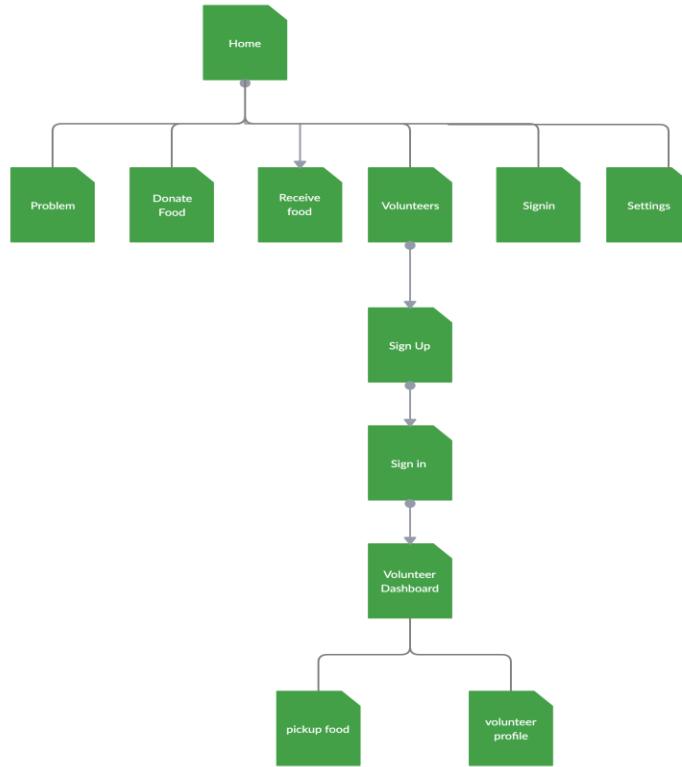
PICK UP

RESTAURANT	DIN TAI FUNG
MENU	CHICKEN ROLL
CUISINE	CHINESE
PICK UP ADDRESS	3450 San Ramon Rd Dublin 94568
PICK UP TIME	04/05/2020 10:00PM
SPECIAL INSTRUCTIONS	CURBSIDE PICKUP
SHELTER	GOODWILL AMBASSADOR
STATUS	CLAIMED

CANCEL      PICKUP CONFIRM

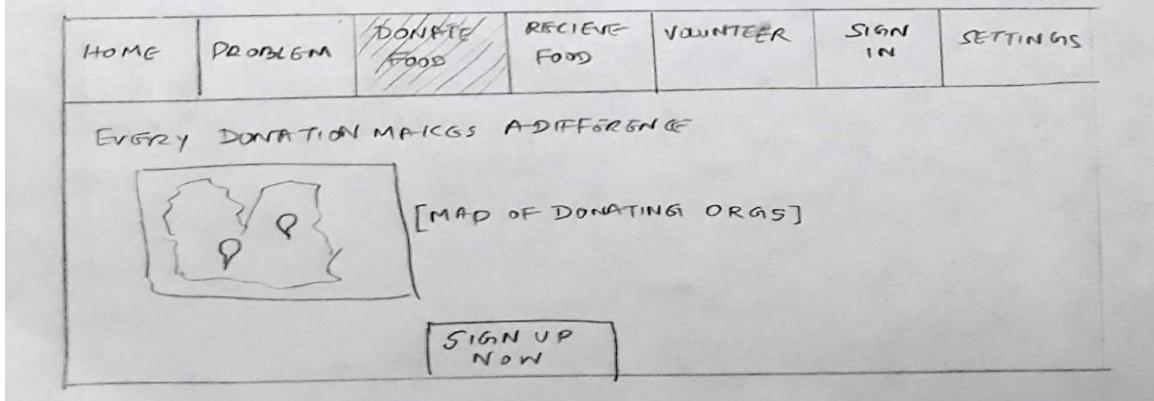
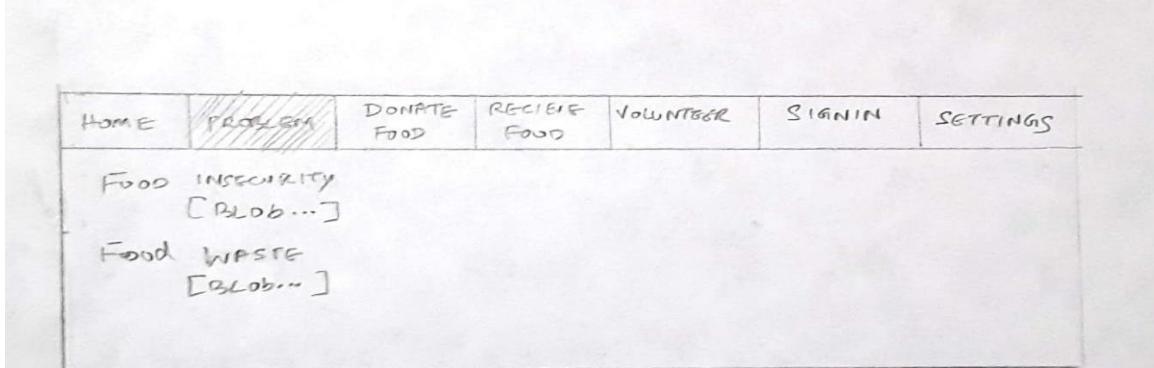
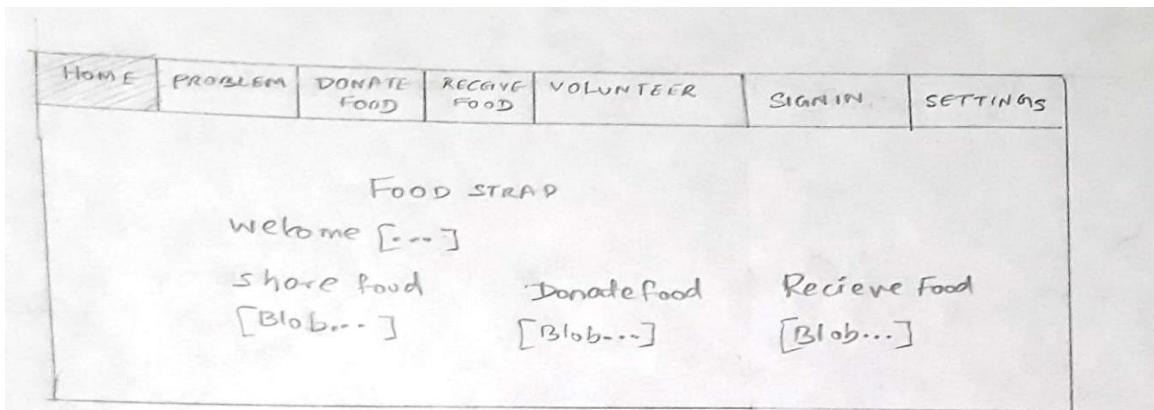
This is the pickup page . It will display the details of the donation that the volunteer wants to pickup . When volunteer clicks pickup confirm , the status is updated and the page navigates back to volunteers' dashboard

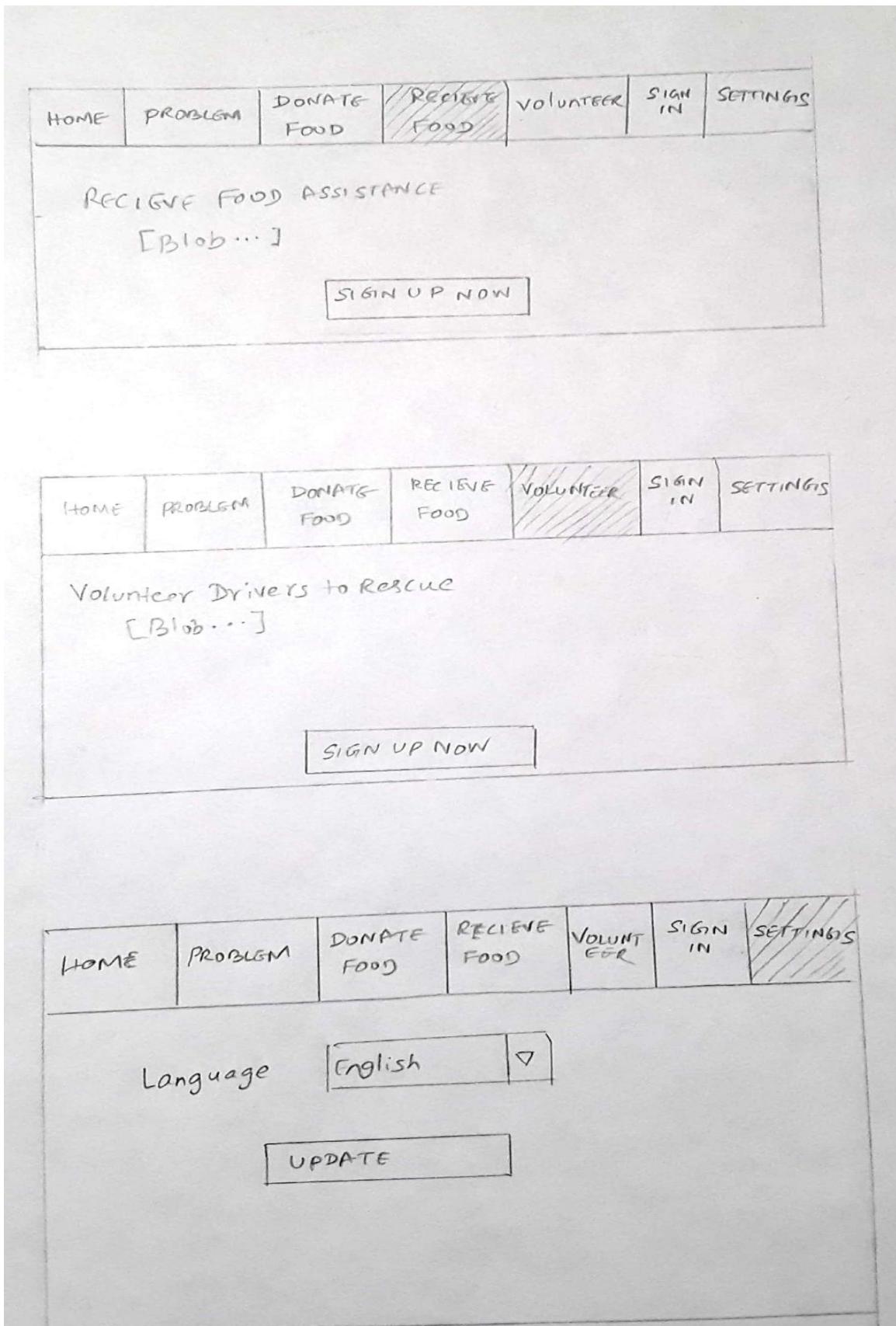
## GUI FLOW FOR VOLUNTEER



### 03. Sketches

The next step in Web UI Design is to design / sketch the basic layouts of individual pages that are going to be in the application. The layout will have the UI components for all the pages. The main thing to note is that the layout in the sketch is not the final design. The sketching process is a way to get things started and receive feedback. The sketches for the application developed as part of this project are below.





**RESTAURANT**

**SIGNUP PAGE**

HOME	PROBLEM	DONATE FOOD	RECEIVE FOOD	VOLUNTEER	SIGN IN	SETTINGS
<input type="checkbox"/> RESTAURANT ▾						
USER NAME		<input type="text"/>				
PASSWORD		<input type="text"/>				
RESTAURANT NAME		<input type="text"/>				
CUISINE		<input type="text"/>				
EMAIL		<input type="text"/>				
ADDRESS			CITY		<input type="text"/>	
STATE			ZIP		<input type="text"/>	
<input type="button" value="SIGN UP"/>						

**SHELTER SIGN UP PAGE**

HOME	PROBLEM	DONATE FOOD	RECEIVE FOOD	VOLUNTEER	SIGN IN	SETTINGS
<input type="checkbox"/> SHELTER ▾						
USERNAME		<input type="text"/>				
PASSWORD		<input type="text"/>				
SHELTER NAME		<input type="text"/>			EMAIL	
ADDRESS		<input type="text"/>				
CITY		<input type="text"/>		STATE	<input type="text"/>	ZIP <input type="text"/>
<input type="button" value="SIGNUP"/>						

**VOLUNTEER SIGN UP PAGE**

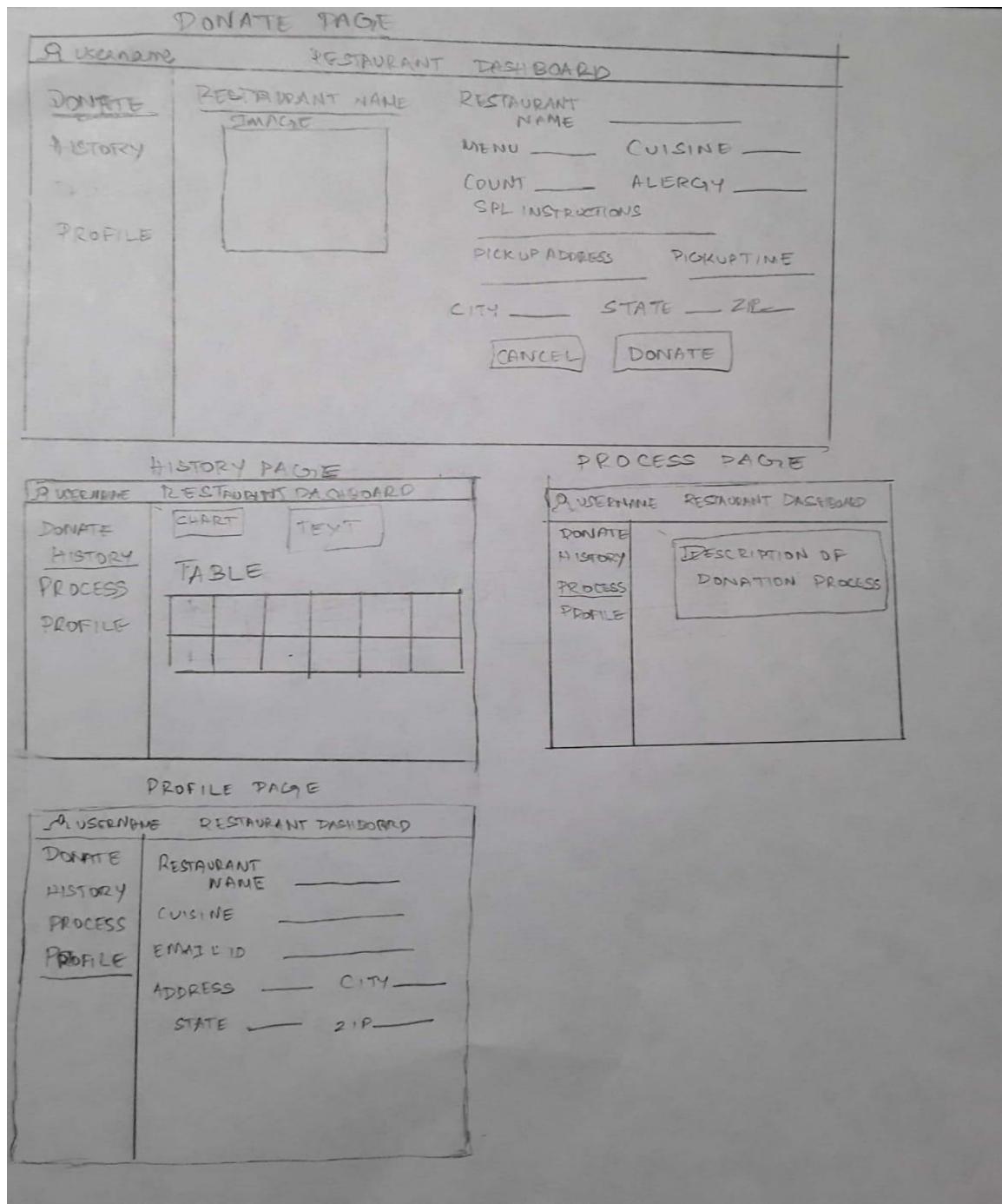
Logo	Home	Problem	Donate Food	Receive Food	Volunteer	Sign In	Settings
<input checked="" type="checkbox"/> Volunteer Username <input type="text"/> Password <input type="password"/> Name <input type="text"/> Email <input type="text"/> <input type="button" value="SIGN UP"/>							

**SIGN IN PAGE**

Logo	Home	Problem	Donate Food	Receive Food	Volunteer	Sign In	Settings
<b>SIGN IN</b> User Name <input type="text" value="Restaurant_Shelter_101"/> Password <input type="password" value="*****"/> User Type <input type="radio"/> Restaurant <span style="float: right;">Restaurant, Shelter 2 Volunte</span>							

**RESTAURANT DASHBOARD**

 Username Donate History Process Profile	Image Map	Dashboard related Text <input type="button" value="Donate"/>
--	--------------	---



**SHELTER DASHBOARD PAGE**

User Name	SHELTER DASHBOARD	
CLAIM HISTORY PROFILE	LOCATION MAP	TEXT

**CLAIM PAGE**

User Name	SHELTER DASHBOARD	
CLAIM HISTORY PROFILE	TABLE OF OPEN DONATIONS	
	CLAIM	CLAIM
	CLAIM	CLAIM
	CLAIM	CLAIM

**CONFIRM CLAIM PAGE**

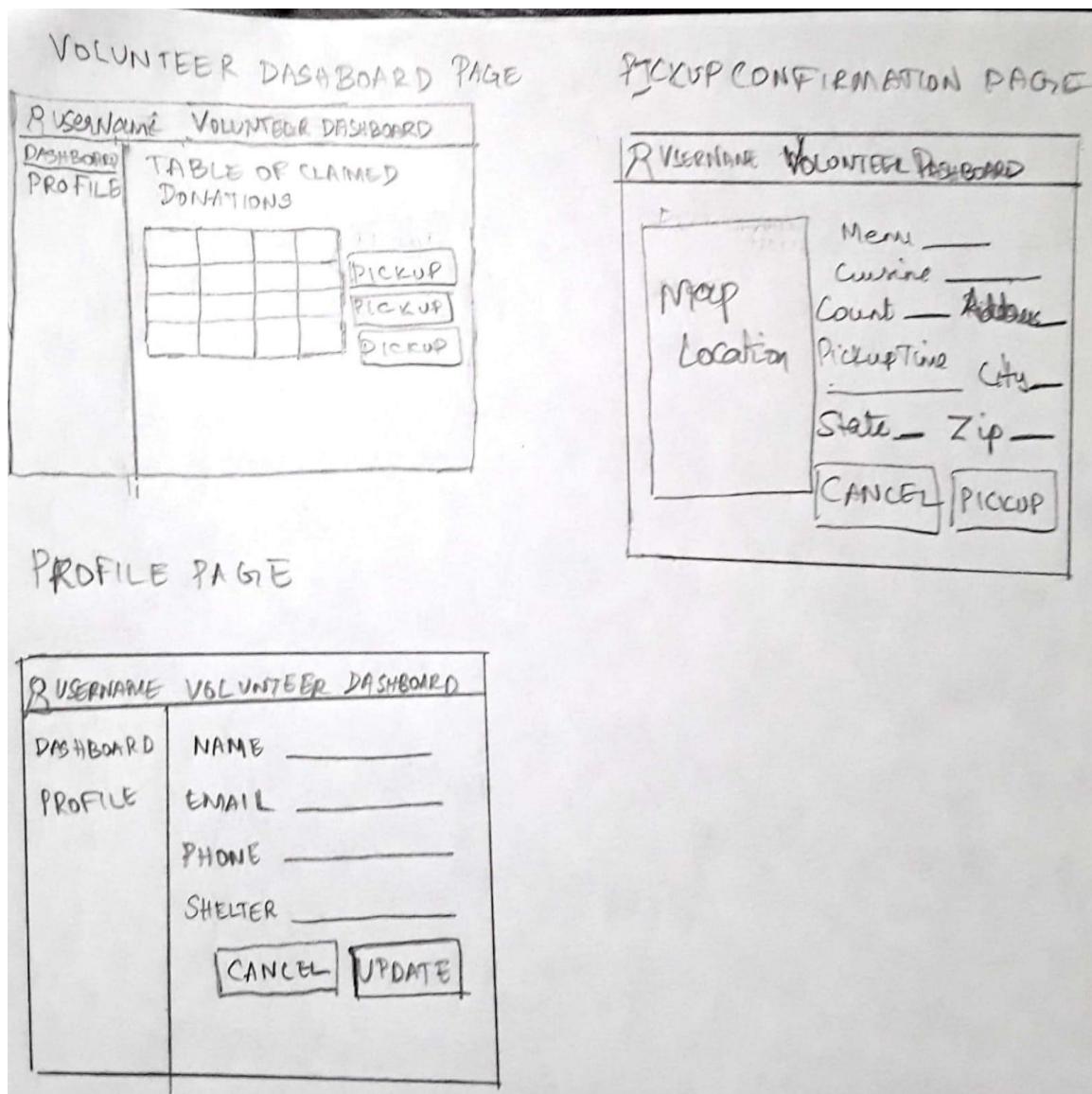
User Name	SHELTER DASHBOARD	
CLAIM HISTORY PROFILE	RESTAURANT NAME <u>Din Tai Fung</u> MENU <u>chicken</u> CUISINE <u>Chinese</u> COUNT <u>20</u> PICKUP <u>Domestic</u> ADDRESS PICKUP <u>05/05/2020 10:00AM</u> TIME ALERLYP <u>Local delivery only</u> CITY <u>DUBLIN</u> STATE <u>CA</u> ZIP <u>94568</u>	
	<input type="button" value="CANCEL"/> <input type="button" value="CONFIRM CLAIM"/>	

**History Page**

User Name	SHELTER DASHBOARD	
CLAIM HISTORY PROFILE	CHART	TEXT
	TABLE OF PAST DONATIONS CLAIMED	

**Profile Page**

User Name	SHELTER DASHBOARD	
CLAIM HISTORY PROFILE	SHELTER NAME _____ EMAIL _____ PHONE _____ ADDRESS _____ CITY _____ STATE _____ ZIP _____ <input type="button" value="CANCEL"/> <input type="button" value="UPDATE"/>	



## 04. Wireframes

Wireframes focus on defining the structure and content of each page. The focus is on identifying what elements to show on the screen and where to place those elements. The wireframes for the pages in our application are illustrated in the below images.

LANDING PAGE

PROBLEM PAGE

FoodStrap

HOME THE PROBLEM DONATE FOOD RECEIVE FOOD VOLUNTEER SIGN IN SETTINGS



Food Insecurity



Food Waste

Legend: █ Object 1 █ Object 2 █ Object 3 █ Object 4 █ Object 5

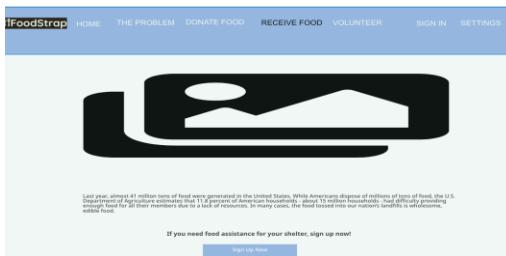
Object	Value
Object 1	10.00
Object 2	9.50
Object 3	8.00
Object 4	7.50
Object 5	7.00

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy enim tempor invidunt at labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clifka kaud gubernari, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy enim tempor invidunt at labore et dolore magna aliquyam erat, sed diam.

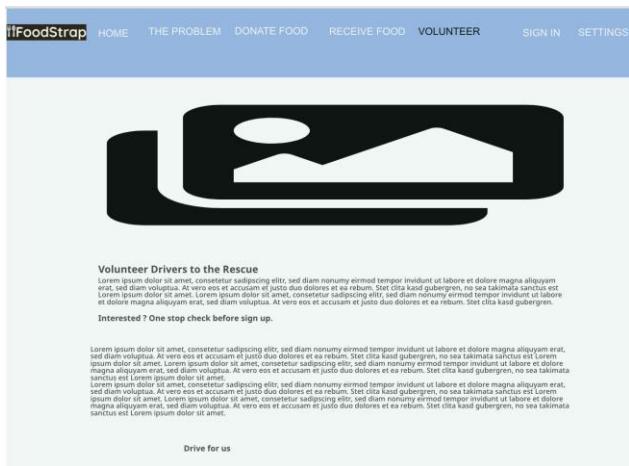
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy enim tempor invidunt at labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clifka kaud gubernari, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy enim tempor invidunt at labore et dolore magna aliquyam erat, sed diam.

#### **DONATE FOOD PAGE:**

RECEIVE FOOD PAGE

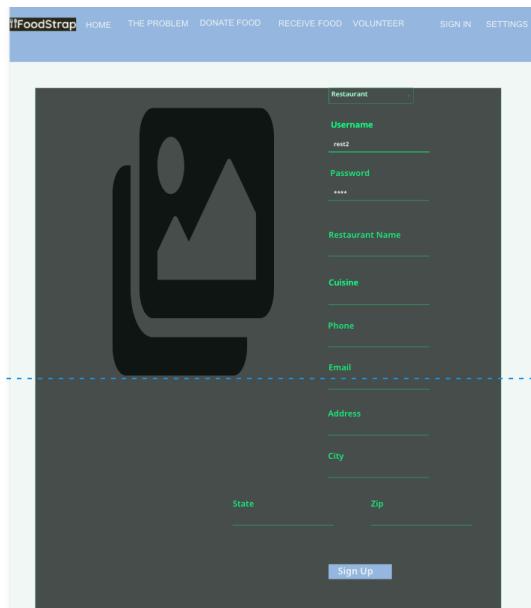


## VOLUNTEER



## SIGNUP

## - RESTAURANT

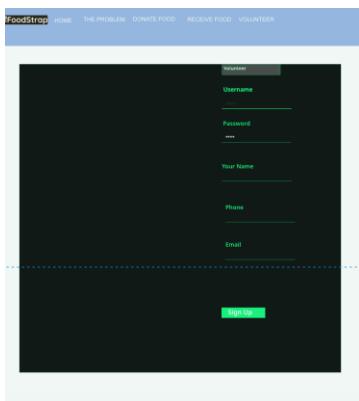


- SHELTER



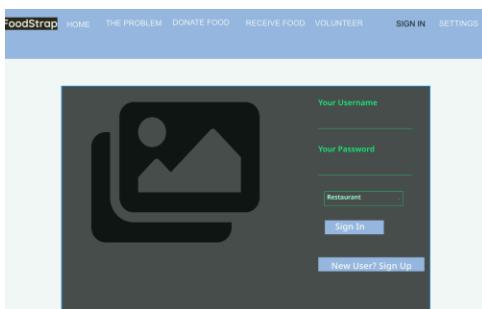
The screenshot shows the 'Shelter' sign-up form. It includes fields for Username, Password, Shelter Name, Phone, Email, Address, City, State, Zip, and a 'Sign Up' button.

- VOLUNTEER



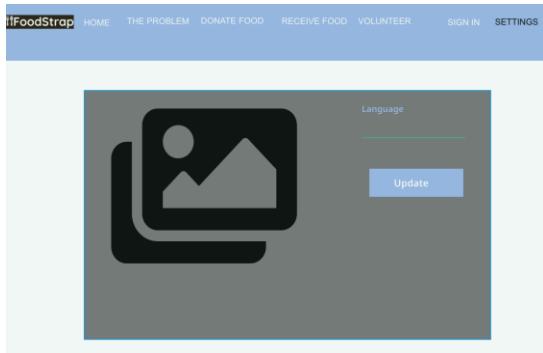
The screenshot shows the 'Volunteer' sign-up form. It includes fields for Username, Password, Your Name, Phone, Email, and a 'Sign Up' button.

SIGN IN



The screenshot shows the sign-in form. It includes fields for Your Username, Your Password, and a Restaurant dropdown menu, along with a 'Sign In' button and a 'New User? Sign Up' link.

## SETTINGS



## RESTAURANT DASHBOARD

A screenshot of the Restaurant Manager Dashboard. The header says "Hello rest2" and "Restaurant Manager Dashboard". On the left, a sidebar has links for Dashboard (which is bolded), Donate, History, Process, and Profile. In the center, there's a map of the world with the United States highlighted in green. To the right of the map is a block of text about food waste in the U.S. and a blue button that says "Donate today and save lives".

## DONATE RESTAURANT DASHBOARD

Hello rest2

### Restaurant Manager Dashboard

[Dashboard](#)  
[Donate](#)  
[History](#)  
[Process](#)  
[Profile](#)

**Menu**

Cuisine	Portion Count

**Allergy Instructions**

**Special Instructions**

**Pick Up Time**

**Pick Up Address**

**City** **State** **Zip**

**Donate**

### HISTORY -RESTAURANT DASHBOARD

Hello rest2

### Restaurant Manager Dashboard

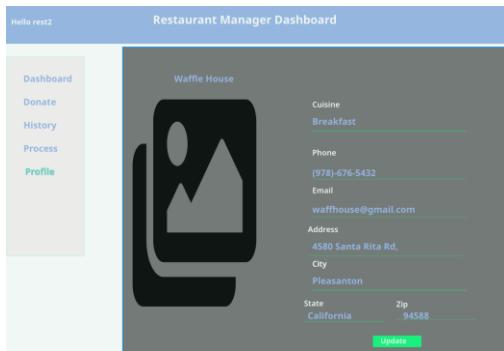
[Dashboard](#)  
[Donate](#)  
[History](#)  
[Process](#)  
[Profile](#)

Menu	Cuisine	Portions	Pickup Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
Pizza	Italian	20	4/11/2020	123.San Jose.94511	Dairy	Pick up on time	Home for Good	Patrick Jane	DONE
Taco	Mexican	20	4/12/2020	945.San Jose.94511	Dairy	Pick up on time	Home for Good	Patrick Jane	DONE

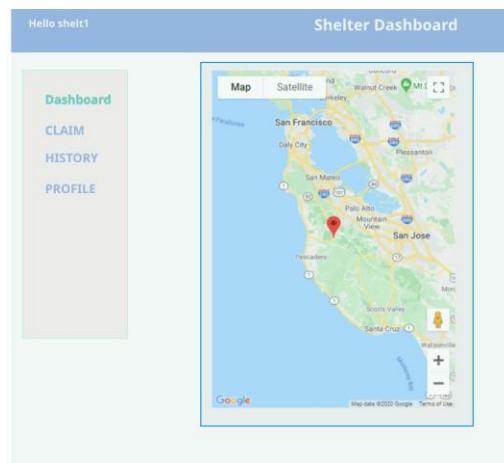
### PROCESS -RESTAURANT DASHBOARD



### PROFILE -RESTAURANT DASHBOARD



## SHELTER DASHBOARD



## CLAIM - SHELTER DASHBOARD



## CONFIRM CLAIM -SHELTER DASHBOARD

**Hello shelt1**

### Shelter Dashboard

**CLAIM**

Dashboard	<b>CLAIM</b>
HISTORY	
PROFILE	

**Menu**  
taco

Cuisine	Portion Count
mex	20

Allergy Instructions  
nuts

Special Instructions  
come

Pick up time  
04/14/2020 12:00 AM

Pick up address  
4850 Santa Rita Rd,

Shelter	Status
Home for Good	OPEN

**CANCEL** **CLAIM**

## HISTORY - SHELTER DASHBOARD

**Hello shelt1**

### Shelter Dashboard

Menu	Cuisine	Portions	Pickup Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
Pizza	Italian	20	4/11/2020	123 San Jose, 94511	Dairy	Pick up on time	Home for Good	Patrick Jane	DONE
Taco	Mexican	20	4/12/2020	945 San Jose, 94511	Dairy	Pick up on time	Home for Good	Patrick Jane	DONE

## PROFILE - SHELTER DASHBOARD

**Hello rest2**

### Shelter Dashboard

**CLAIM**

Dashboard	<b>CLAIM</b>
History	
Profile	

**Shelter Name**  
Home For Good

**Phone**  
(978)-676-5432

**Email**  
home4good@gmail.com

**Address**  
4590 artist common walk

**City**  
Fremont

**State**  
California

**Zip**  
94536

**Update**

## VOLUNTEER DASHBOARD

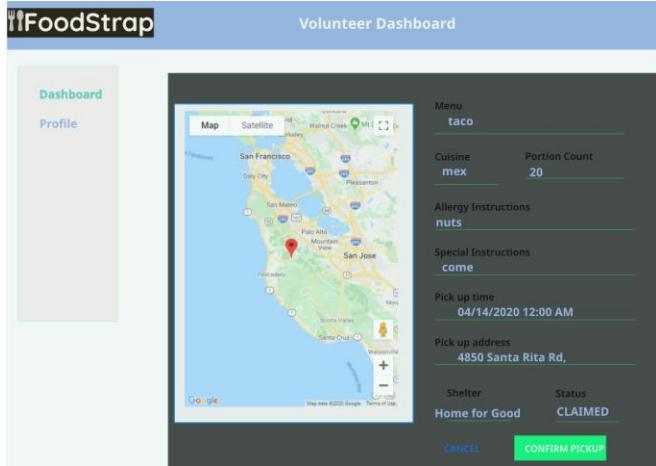
**FoodStrap**

### Volunteer Dashboard

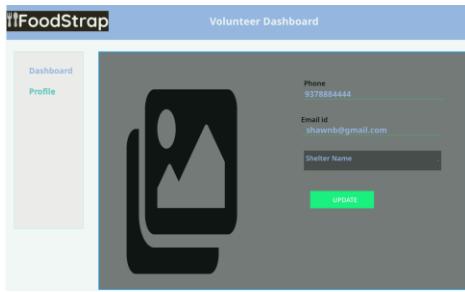
**CLAIM**

Menu	Cuisine	Portions	Pickup Date Time	Address	Allergy Information	Special Instructions	Shelter	Status
TACO	MEXICAN	20	4/11/2020 9:00 PM	789,SAN JOSE 94512	CONTAINS GLUTEN	PICK UP ON TIME	HOME FOR GOOD	<b>CLAIMED</b> <b>PICKUP</b>

## PICKUP - VOLUNTEER DASHBOARD



## PROFILE-VOLUNTEER DASHBOARD



## 05. Application Features

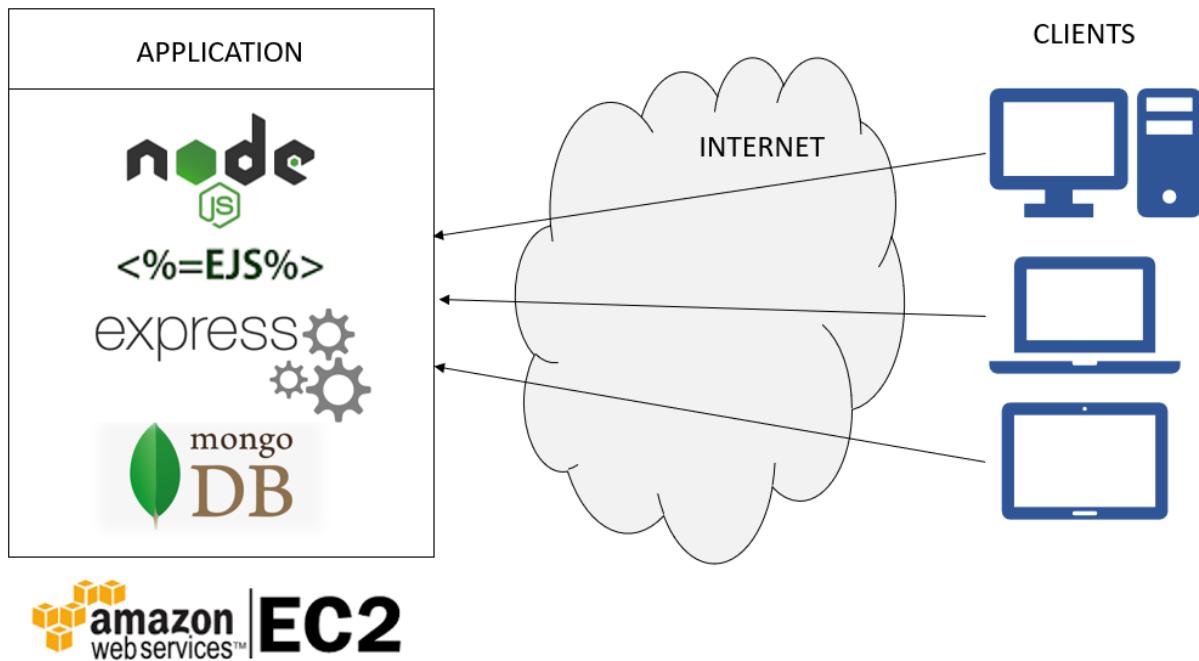
The pages in the application for various features are

- Landing page of App
- Problem page
- Donations page
- Shelters page
- Volunteers page
- Settings page
- Sign Up Page
- Sign In Page
- Restaurant Dashboard Page
- Restaurant Donate Page
- Restaurant Donations History Page
- Restaurant Donation Process Page
- Restaurant Profile Page
- Shelter Dashboard Page
- Shelter-Claim Donation Page

- Shelter - History of past claims Page
- Shelter -Profile Page
- Volunteer Dashboard Page
- Volunteer Pickup Page
- Volunteer Profile Page
- Log Out

## 7. High Level Architecture Design

The application is developed in NodeJS using Express and EJS frameworks and MongoDB is the database. The application is hosted in AWS EC2 instance.



### 01. Technology Stack

#### Front end

- NodeJS EJS Embedded JavaScript
- HTML5
- CSS3
- Library: JQuery, BootStrap, D3JS, Google maps, Google Charts

#### Back end

- NodeJS Express.js
- Library: mongod, node-geocoder, node-fetch, http-errors, debug, morgan, client-sessions, cookie-parser

#### Database

- MongoDB

#### Cloud Deployment

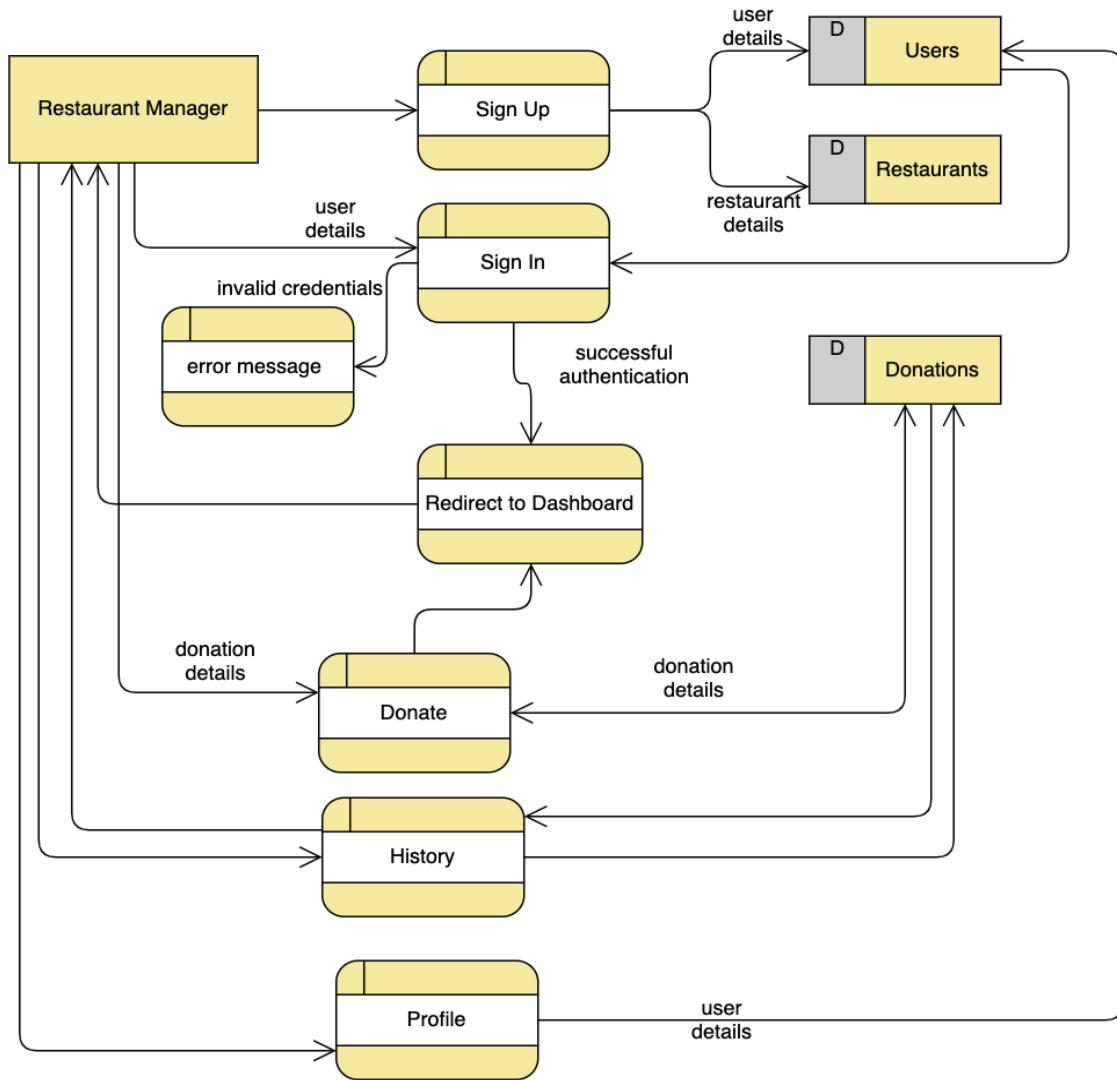
- AWS EC2

#### Test Automation

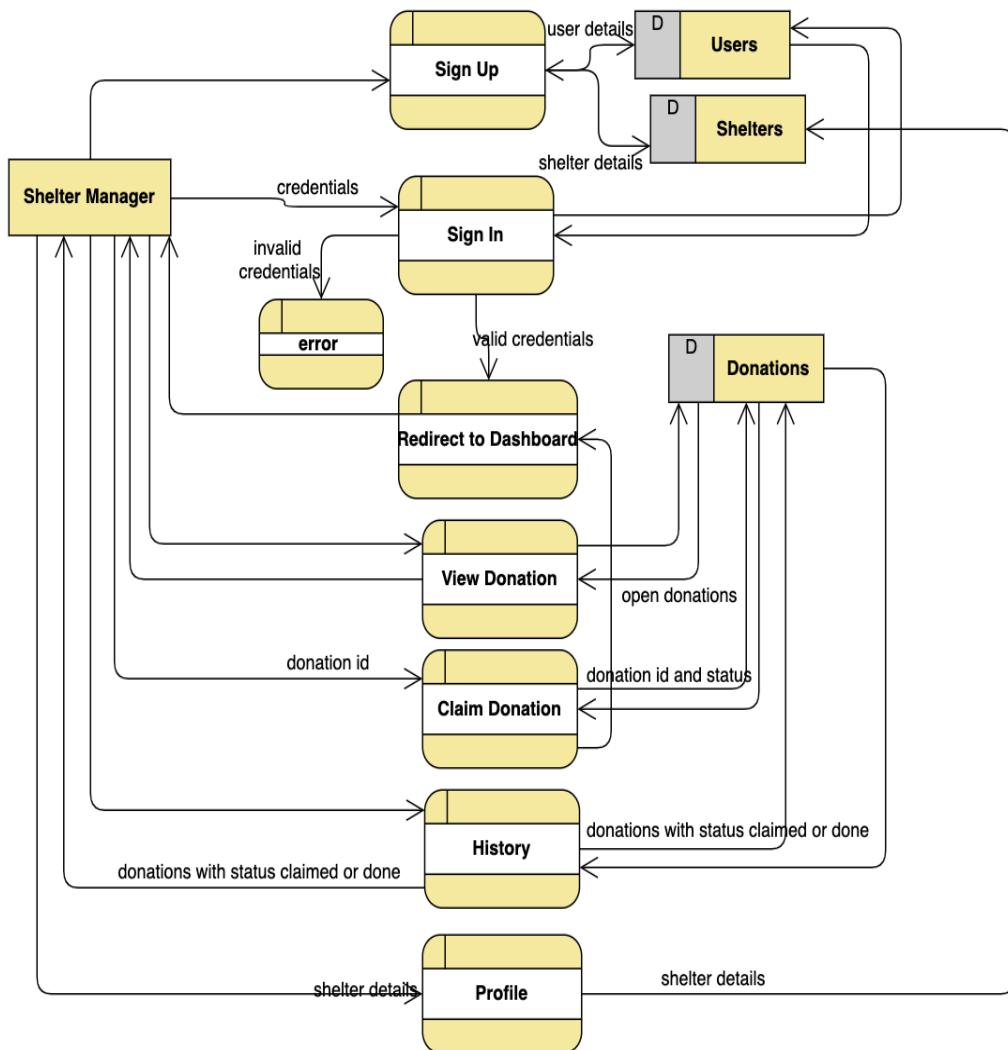
- Mocha
  - NodeJS
  - Library: Chai, Cheerio, Request
- Selenium
  - Selenium-webdriver
  - Chrome driver

## 8. Data Flow Diagram

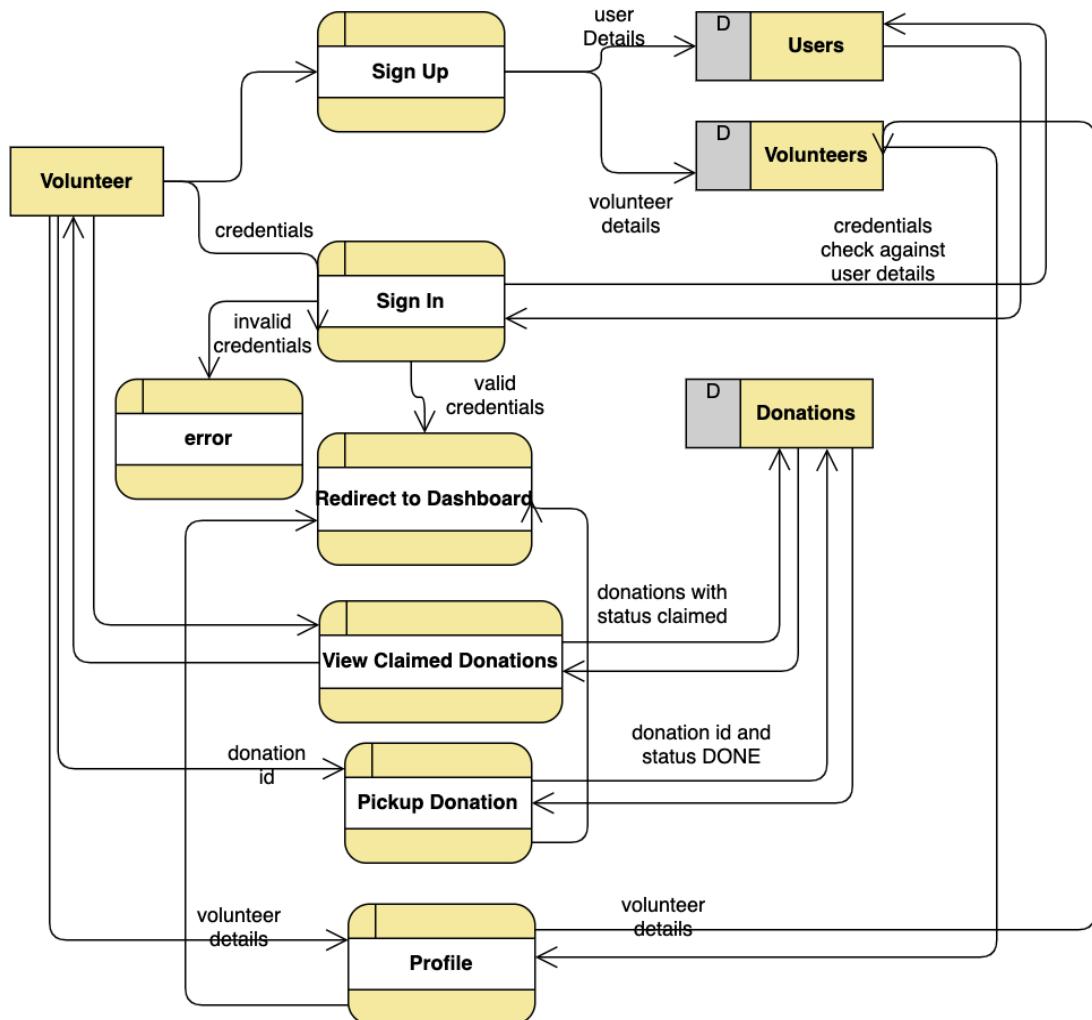
## 01. Restaurant Manager



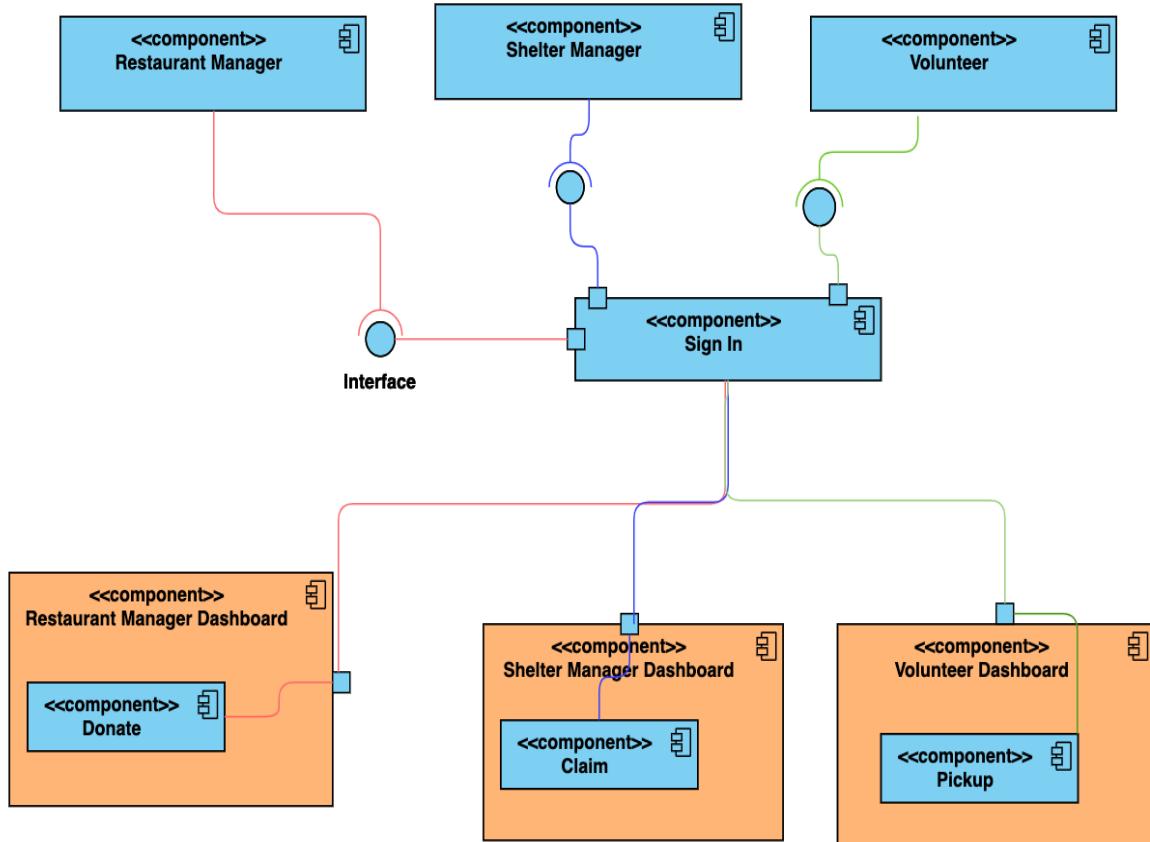
## 02. Shelter Manager



### 03. Volunteer



## 9. Component Diagram

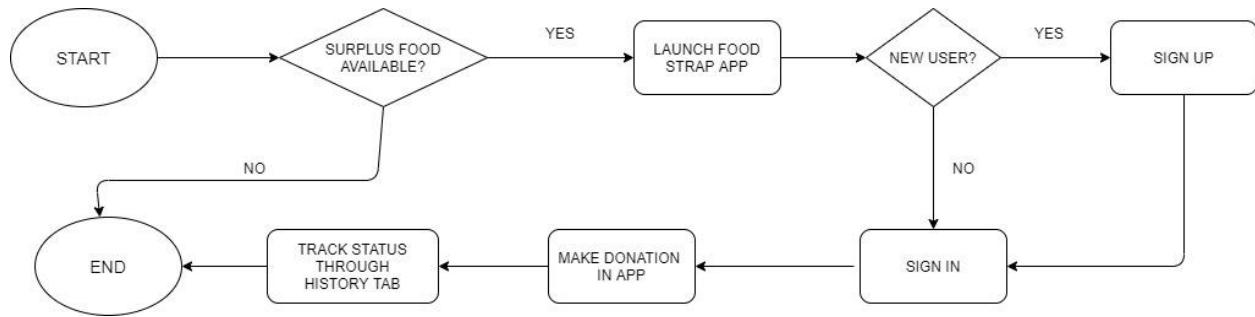


## 10. Workflows

The workflows for different users are shown below.

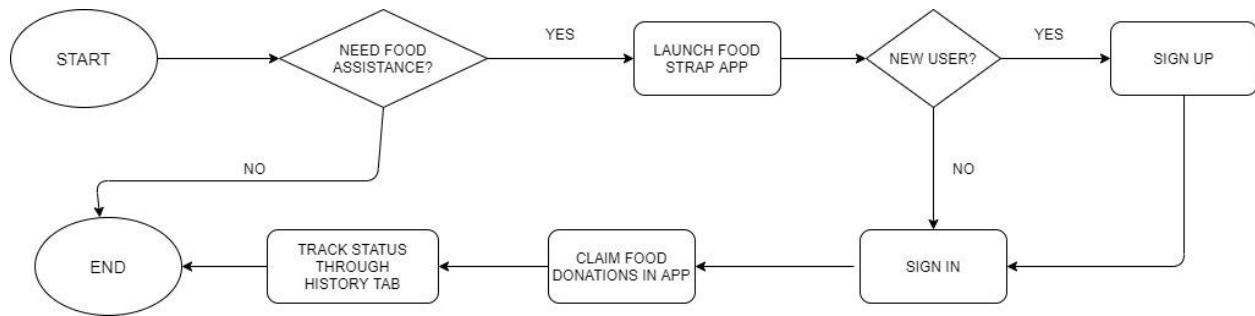
### 01. Restaurant

1. Restaurant Manager registers himself in the app.
2. If surplus food is available, he makes a donation in the app.
3. Verify donation process through Process tab
4. Tracks status through History Tab
5. Update profile of restaurant through Profile tab



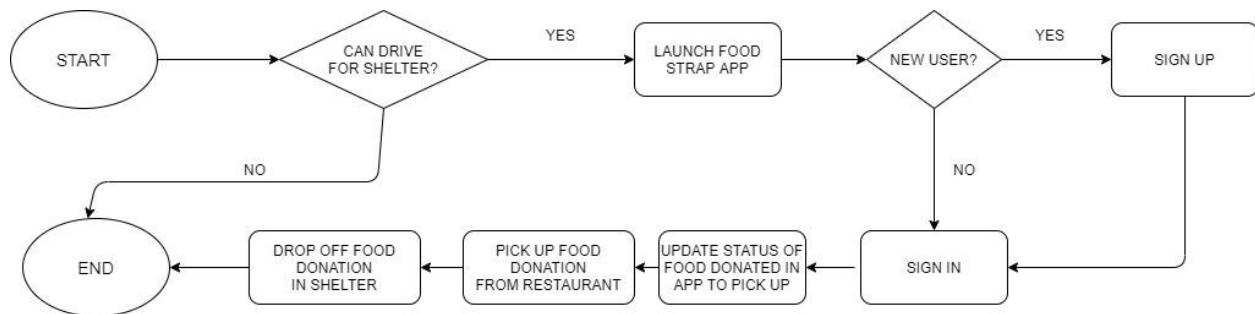
## 02. Shelter

1. Shelter manager registers himself in the app.
2. If the shelter needs food assistance, he claims donated food from the app.
3. Tracks status through Claims History tab
4. Update profile of shelter through Profile tab



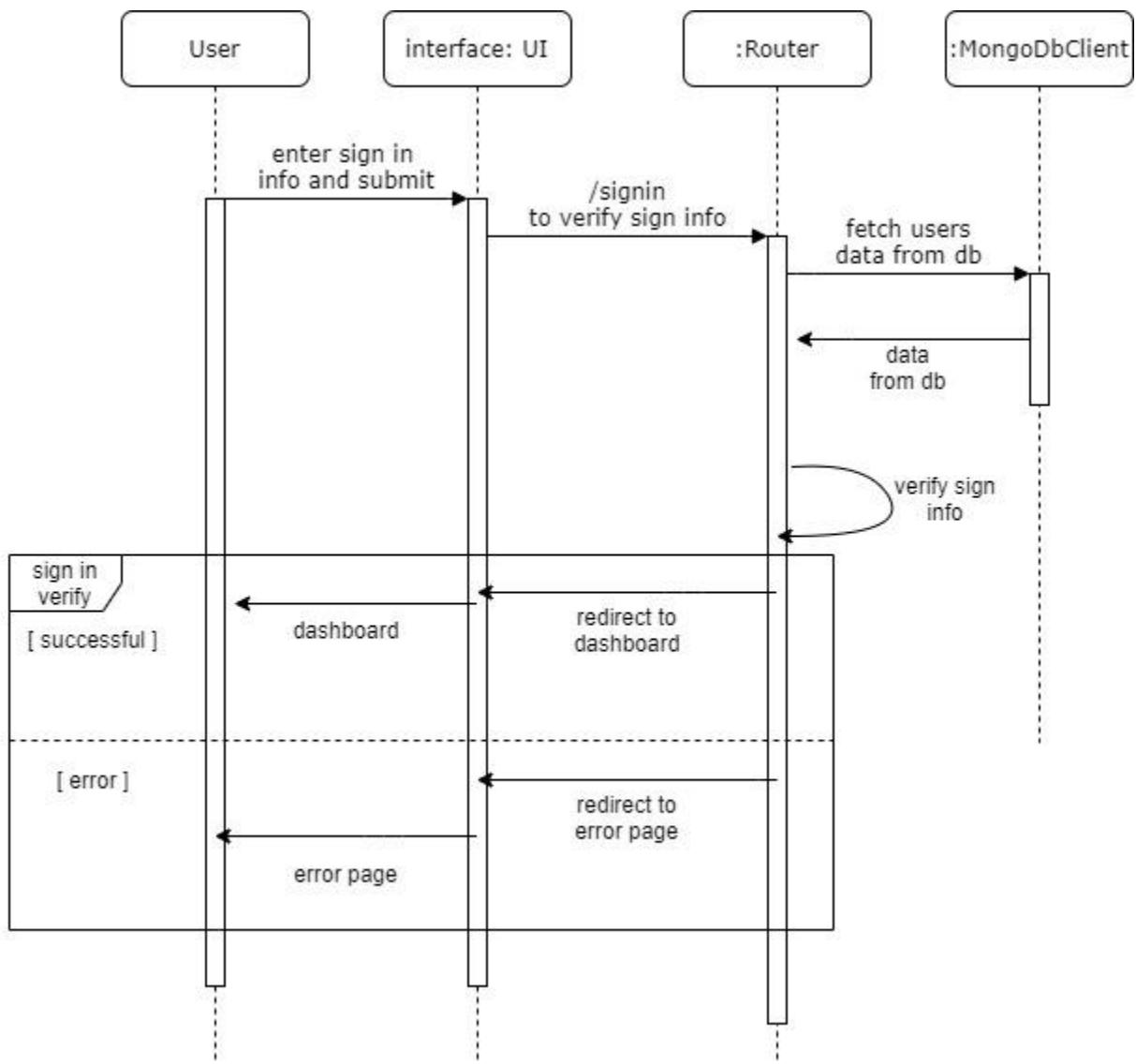
## 03. Volunteer

1. A volunteer for the shelter registers himself in the app.
2. He checks the app for claimed donations and picks up them.
3. Update profile of volunteer through Profile tab

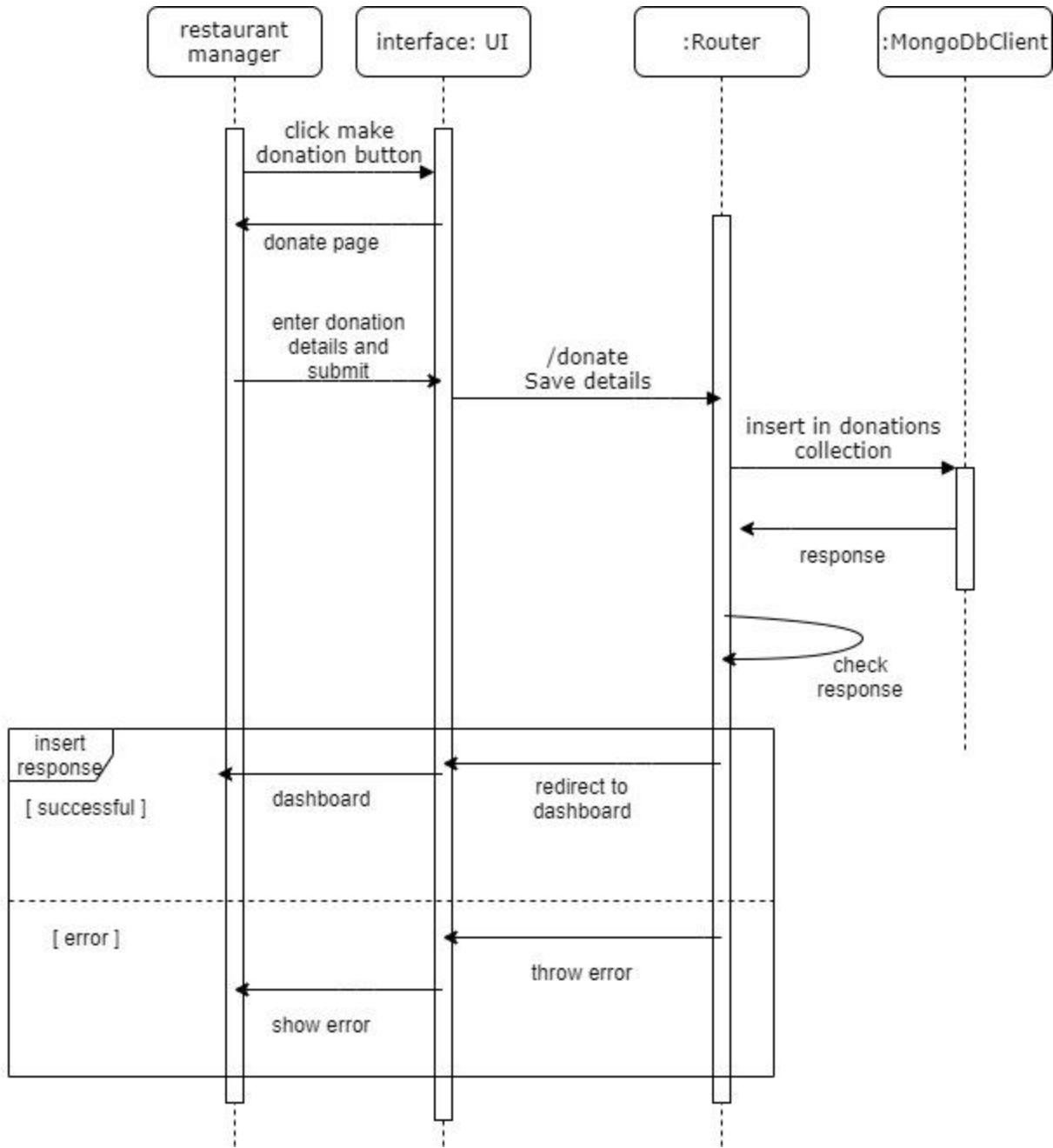


## 11. Sequence Diagram

### Sign In flow of user



### Making Donation flow of restaurant manager



## 12. HTML5 Features

### 1. Header

Header tag is used to define a header section for pages or elements in the application. For heading and images required in the header area, this tag is used. E.g.: In the Problems page, the header tag is used, and it contains the image which is required to be displayed as a banner in the header.

Sample code:

```
<header align="center"></header>
```



Header section  
Of container  
div element

## 2. Footer

Footer tag is used to display information in the footer area of pages or sections. It is used to display the copyright information in the footer area.

Sample code:

```
<footer class="footer mt-auto py-3 bggreyfooter verticalAlignCenter">
<div class="container verticalAlignCenter topmarginFooter">
<br>©
<script>
document.write(new Date().getFullYear())
</script> Created by Parvathy & Mayura for CMPE280 Web UI Design
</div>
</footer>
```

### From where do we get our food?

They have few sources from where we get the food. It can be restaurants who have surplus food, caterers who have leftover food, individual donations, local businesses around the area.

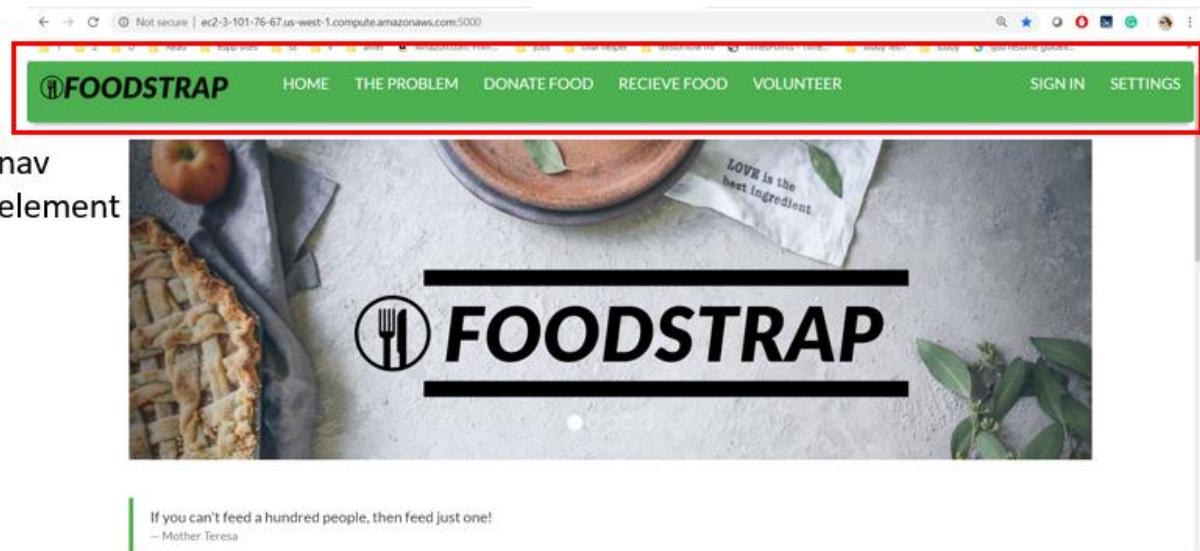
© 2020 Created by Parvathy & Mayura for CMPE280 Web UI Design

## 3. Nav

The <nav> tag defines a set of navigation links. It is used to define navigation links in the header area in all pages of the application. In the initial pages, before login, it is used to display links for home, problem, donors, shelters, volunteers, sign in and settings pages. Once login is done, nav tag is used to display links in the dashboard. In the dashboard, it contains a menu icon, page heading and logo.

Sample code:

```
<nav class="navbar navbar-default">
<div class="container-fluid">
<ul class="nav navbar-nav">
<li><a class="navbar-brand" href="/"></a></li>
<li><a href="/"><%= navLabels.home %></a></li>
<li><a href="/problem"><%= navLabels.problem %></a></li>
<li><a href="/donors"><%= navLabels.donate %></a></li>
<li class="active"><a href="/shelters"><%= navLabels.shelters %></a></li>
<li><a href="/volunteers"><%= navLabels.volunteer %></a></li>
</ul>
<ul class="nav navbar-nav navbar-right">
<li><a href="/signin"><%= navLabels.signin %></a></li>
<li><a href="/setting"><%= navLabels.settings %></a></li>
</ul>
</div>
</nav>
```

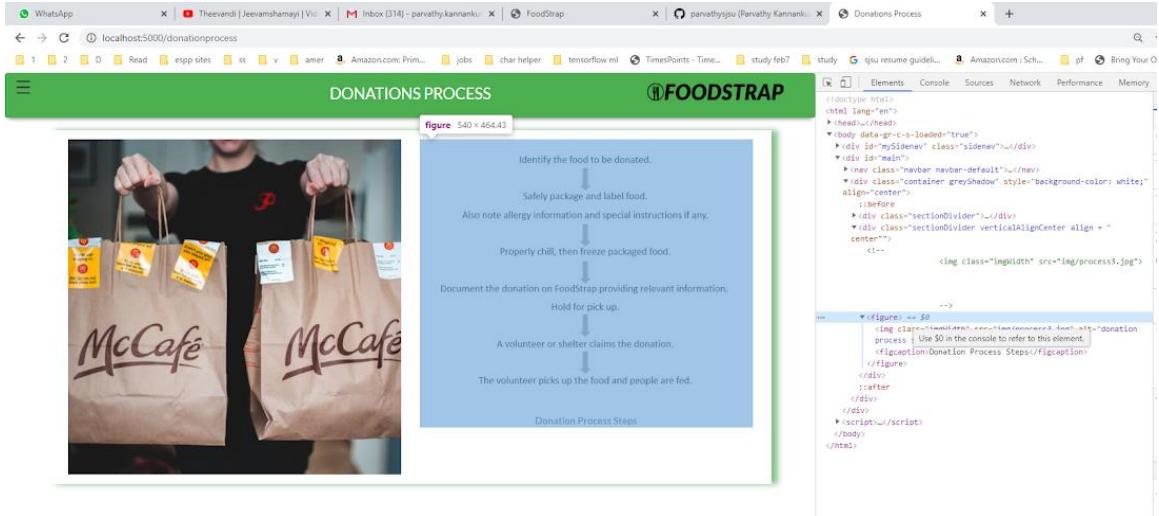


#### 4. Figure

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. It is used in the application to include images.

Sample code:

```
<figure>
  ">
  <figcaption><%= msgs.donProcessCaption %></figcaption>
</figure>
```



## 5. Figcaption

<figcaption> element is used to define a caption for the images used in the app. All the figcaption text is fetched from properties file, for different languages.

Sample code:

```
<figure>
  ">
  <figcaption><%= msgs.donProcessCaption %></figcaption>
</figure>
```

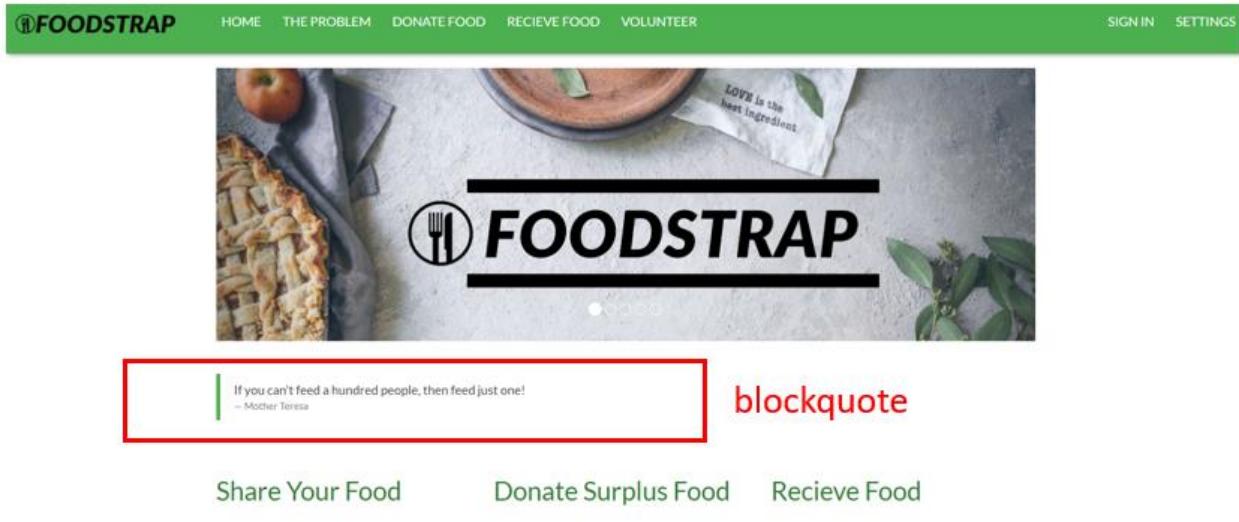


## 6. Blockquote

<blockquote> element is used to define quotes and display it in a style that matches a quote. The quote author is also mentioned in the quote. Quotes by Mahatma Gandhi and Mother Teresa are added in Index, donate food and receive food pages.

Sample Code:

```
<blockquote>
  <p><%= msgs.quote %></p>
  <footer><%= msgs.author %></footer>
</blockquote>
```



## 7. Input type number

Input type of number is used in forms that require a number field. Eg: Number of portions being donated field in donate form is a number and input type number element is used for that.

Sample Code:

```
<input type="number" name="count" id="count" value="" required />
```



DONATE

**Input type number element**

Menu

Cuisine

Allergy Information

Special Instructions

Pick Up Time

Pick Up Address

City      State      Zip

Portion Count  
25

Donate

#### 8. Input type email

Input type of email is used in forms that require an email field. Eg: email of volunteer field in volunteer sign up form is an email and input type email element is used for that.

Sample Code:

```
<input type="email" name="emailid" id="emailid" value="" />
```



The image shows a healthy salad in a black bowl, garnished with various toppings like carrots, onions, and nuts. Overlaid on the top left is the text "SIGN UP TODAY". To the right is a sign-up form with fields for "Volunteer", "Username", "Password", "Your Name", "Your Phone Number", and "Your Email Id" (which contains the value "parvathy.krishnan@sjtu.edu"). A red box highlights the "Your Email Id" field. Below the form is a green "SIGN UP" button.

Volunteer \_\_\_\_\_

Username \_\_\_\_\_

Password \_\_\_\_\_

Your Name \_\_\_\_\_

Your Phone Number \_\_\_\_\_

Your Email Id  
parvathy.krishnan@sjtu.edu

SIGN UP

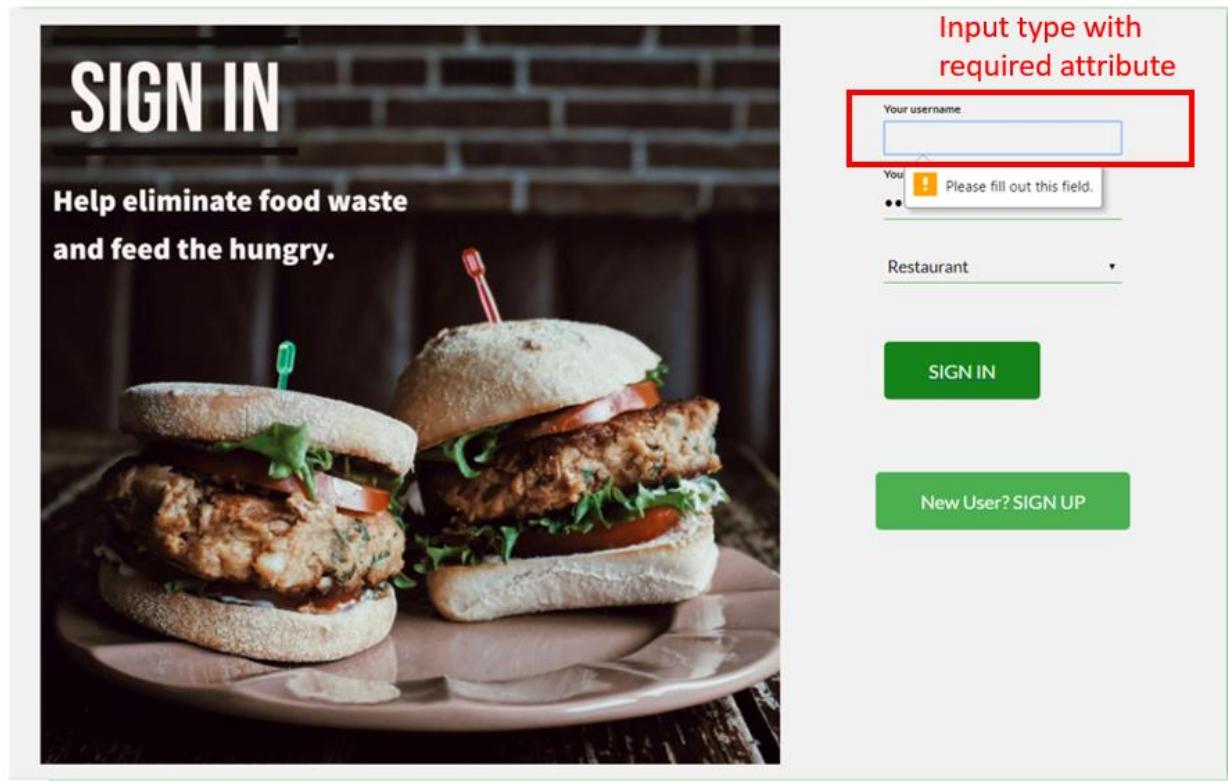
**Input type  
email element**

#### 9. Required attribute

Certain fields in forms are required to be filled like username in sign in form. Required attribute is used for those input elements.

Sample Code:

```
<input type="text" name="username" id="username" value="" required />
```



## 13. Server-Side Design

### 01. APIs

The APIs used in the application are described below. It includes both backend APIs and page rendering APIs for back end and front end respectively.

#	ENDPOINT	METHOD	DESCRIPTION
1.	/	GET	To render the index page. It also fetches db data for displaying in UI
2.	/error_msg	GET	To render sign in incorrect login error page. Displays error message.
3.	/restList	GET	Returns the restaurant list. This is then used by another API to display the restaurants in google map.
4.	/donPosition	GET	Returns the donation pick up location. This is then used by another API to display the location in google map.

5.	/shelterPosition	GET	Returns the shelter location. This is then used by another API to display the location in google map
6.	/signin	GET	To render the sign in page
7.	/signup	GET	To render the sign-up page
8.	/setting	GET	To render the setting page
9.	/setting	POST	To save the language updated in the Settings page. This value is then used to display all pages in the app
10.	/donate	GET	To render the donate page
11.	/donationshistory	GET	To render donations history page for a restaurant manager after fetching donations records from db
12.	/restaurant_dashboard	GET	To render the restaurant dashboard page for a restaurant manager after fetching data from db
13.	/donationprocess	GET	To render the donation process page for a restaurant manager.
14.	/pickup/:donid	GET	To render the pick page for a donation for a volunteer after fetching donation data from db. Volunteer confirms the pickup or cancels in this page.
15.	/claim/:donid	GET	To render the claim page for a donation for a shelter after fetching donation data from db. Shelter manager confirms the claim or cancels in this page
16.	/volunteer_dashboard	GET	To render the volunteer dashboard page for a volunteer after fetching data from db
17.	/pickupdonation	POST	To update the status of donation in db to DONE after being picked up by a volunteer
18.	/claimdonation	POST	To update the status of donation in db to CLAIMED after being claimed up by a shelter
19.	/shelter_dashboard	GET	To render the shelter dashboard page for a shelter manager after fetching data from db

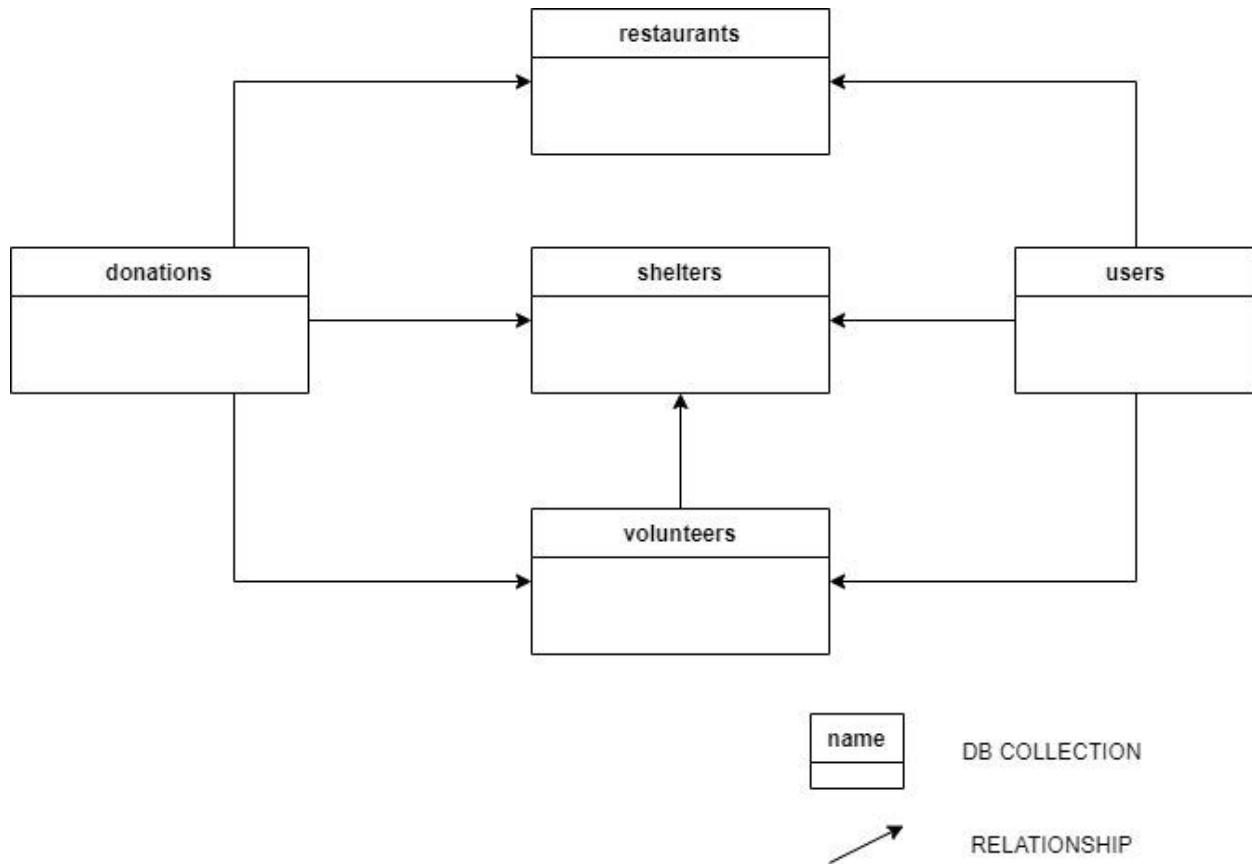
20.	/donate	POST	To save donation data after a restaurant manager makes a donation.
21.	/signup	POST	To save new user details after a new user signs up in the app
22.	/signin	POST	To check sign in values for login. If successful redirect to dashboard and error page for incorrect login.
23.	/problem	GET	To render the problem page. It explains the problem of food waste
24.	/shelter_claim	GET	To render the claims page for a shelter after fetching data from db. It displays all the donations that are available for claim
25.	/past_claim	GET	To render the claims history page for a shelter after fetching data from db.
26.	/donors	GET	To render donate food page. It displays all the current donor restaurants
27.	/shelters	GET	To render receive food page. Here shelters can sign up
28.	/volunteers	GET	To render volunteer page. Here volunteers can sign up
29.	/vol_profile	GET	To render the profile of a volunteer. Volunteer can see and update his profile from this page
30.	/rest_profile	GET	To render the profile of a restaurant. Restaurant manager can see and update the profile from this page
31.	/rest_profile	POST	To save the updated profile of restaurant manager user.
32.	/shelter_profile	GET	To render the profile of a shelter. Shelter manager can see and update the profile from this page
33.	/shelter_profile	POST	To save the updated profile of shelter manager user.
34.	/vol_profile	POST	To save the updated profile of a volunteer user.

## 02. DB Design

MongoDB is the database used to store data. Using it we can easily store data and possible to add additional attributes later. It scales out easily.

Collections used are:

- restaurants
- shelters
- volunteers
- donations
- users



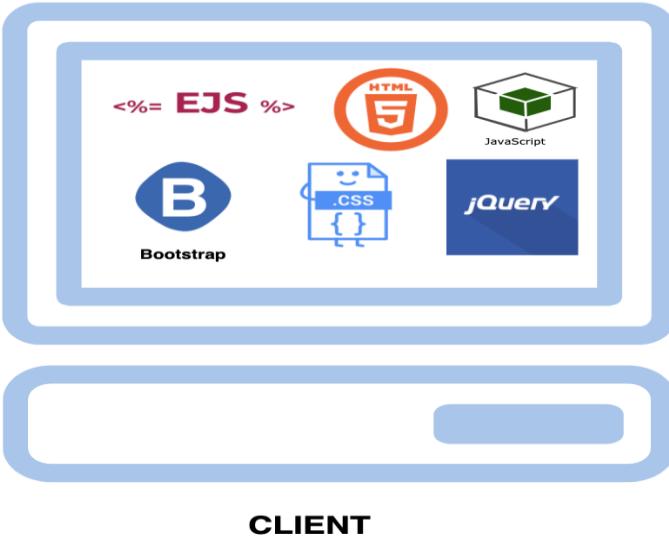
Collection	Description	Sample record
restaurants	This collection stores the details of a restaurant like name, address, cuisine	{         "_id": {             "\$oid": "5e9a449c0293bd05a40900e9"         },         "username": "rest4",         "name": "Dominos",         "cuisine": "Pizza",         "phone": "123456",         "address": "123 Main St, New York, NY 10001"     }

		<pre> "emailid":"abcd@gmail.com", "addr":"N4th street", "city":"San Jose", "state":"CA", "zip":"95113" } </pre>
shelters	This collection stores the details of shelters like name, address.	<pre> {   "_id": {     "\$oid": "5e9389bfa05c215da894b490"   },   "username": "sh1",   "name": "Home For Good",   "phone": "4001521235",   "emailid": "sh1@gmail.com",   "addr": "935",   "city": "San",   "state": "CA",   "zip": "95112" } </pre>
volunteers	This collection stores the details of volunteers like name, phone, email.	<pre> {   "_id": {     "\$oid": "5e9389f7a05c215da894b492"   },   "username": "vol1",   "name": "Pinky susan",   "phone": "4002224444",   "emailid": "pinky@gmail.com",   "shelter": {     "username": "sh1",     "name": "Home For Good"   } } </pre>
users	This collection stores the details of users like username, password and user type	<pre> {   "_id": {     "\$oid": "5e938930a05c215da894b48d"   },   "usertype": "restaurant",   "username": "test",   "password": "test" } </pre>
donations	This collection stores the details of donations like menu, portion count, pick up time, address, restaurant	<pre> {   "_id": {     "\$oid": "5ea9e1bdb4d1b0346c1886f1"   }, } </pre>

	<p>donating, shelter claiming and volunteer pick up.</p> <pre> "restaurant": "rest2", "menu": "salsa and tacos", "cuisine": "Mexican", "count": "50", 'allergy": "nuts", "notes": "abc", "pickuptime": "04/29/2020 7:00 PM", "pickupaddr": "935 N2nd Street", "pickupaddrcity": "San Jose", "pickupaddrstate": "CA", "pickupaddrzip": "95112", "status": "DONE", "shelter": {   "username": "sh1",   "name": "Home For Good" }, "volunteer": {   "username": "vol",   "name": "Shawn Booth" } } </pre>
--	--

## 14. Client-Side Design

Client Side refers to everything in a web application that is displayed or takes place on the client (i.e. end user device). The application has been developed using the MVC pattern. The client side refers to the ‘View’ part of the pattern. The client side in the application utilizes EJS, HTML5, JavaScript, CSS, Bootstrap and jQuery to render the Graphical User Interface.

**EJS:**

EJS is a simple templating language that lets the user generate HTML markup with plain JavaScript. The backend server code is developed in node.js and express. The default view engine of express is jade. EJS is an alternative view engine. The html template files have the extension .ejs files

**HTML5:**

The html template follows the HTML5 format. HTML5 offers various features like header, footer, different input types like number, email, navigation bar etc. To utilize those features and render a rich graphical user interface, HTML5 is a perfect choice. Other features from previous HTML versions such as title tag, image tag, h1 tag have also be used to improve search engine optimization

**BOOTSTRAP:**

Bootstrap is a free and open-source CSS framework focused on responsive front-end web development. It contains CSS and JavaScript based design templates for typography, forms, buttons, navigation and other interface components. For e.g. Bootstrap datetime picker component has been used for selecting the pickup date and time. Another use of bootstrap in our application is for cross-browser/cross-device compatibility Bootstrap supports most of the modern browsers and also supports browsers in mobile devices such as android and iPhone.

**CSS:**

Cascading Style Sheets is a language for styling the presentation of HTML elements. It can also define variations in page layouts. The style definitions for all the pages stored in an external CSS file. CSS plays a very big part in the rendering of the UI.CSS was used for styling the responsive side navigation bars

that is present in all the dashboard pages. CSS is a great help while designing the UI as it provides an option to change background color, or define font-family, change the font size for text elements and more importantly in the alignment of the elements. The landing page of the application has an image carousel feature that was designed using CSS styling. The code snapshot for the carousel is below:

#### HTML

```
<div class="container">
  <div id="myCarousel" class="carousel slide" data-ride="carousel">
    <!-- Indicators -->
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
      <li data-target="#myCarousel" data-slide-to="2"></li>
      <li data-target="#myCarousel" data-slide-to="3"></li>
      <li data-target="#myCarousel" data-slide-to="4"></li>
    </ol>
    <!-- Wrapper for slides -->
    <div class="carousel-inner">

      <div class="item active">
        
      </div>
      <div class="item">
        
      </div>
      <div class="item">
        
      </div>
      <div class="item">
        
      </div>
      <div class="item">
        
      </div>
    </div>
  </div>
```

#### CSS

```
.carousel-indicators li {
  width: 16px;
  height: 16px;
}

.carousel-indicators .active {
  width: 18px;
  height: 18px;
  background-color: #fff;
```

## JAVASCRIPT

JavaScript was used to handle events such as onclick by writing a function that will be triggered when the event happens. In the application, the side navigation bars in the dashboard pages appear only on click of the hamburger icon. Similarly on click of the x the navigation bar is closed. The snapshot of the JavaScript functions responsible for these events is given below.

```
function openNav() {
    document.getElementById("mySidenav").style.width = "250px";
    document.getElementById("main").style.marginLeft = "250px";
}

function closeNav() {
    document.getElementById("mySidenav").style.width = "0";
    document.getElementById("main").style.marginLeft= "0";
}
```

Also, Google maps is rendered using the JavaScript function initMap. Many other JavaScript libraries such as D3.js , jQuery , DataTable are also used for the UI design.

# 15. Test Automation

## 01. Test Automation using Mocha

Mocha, a JavaScript test framework on Node.js was used for asynchronous testing.

Command to run:

1. Start application
  - a. Npm install
  - b. npm start
2. Run the test cases
  - a. npm test

Script is located in folder FoodStrap\foodstrap\test

Flows and pages tested are Sign In, Sign Up, Receive Food page, Settings Page and negative flow for an invalid page. For each page, the status code and page content is verified. Verification is done using the expect method of Chai library. To verify page content, Cheerio library is used. Cheerio implements a

subset of jQuery and it **parses markup and provides an API for traversing/manipulating the resulting data structure.**

```
PS C:\Users\parva\OneDrive\Desktop\sjtu1\spring2020\280\project2020April\FoodStrap\foodstrap> npm test
> foodstrap@0.0.1 test C:\Users\parva\OneDrive\Desktop\sjtu1\spring2020\280\project2020April\FoodStrap\foodstrap
> mocha

Automated test Cases
  SignIn Page Test Cases
    ✓ SignIn page
    ✓ SignIn page content
  SignUp Page Test Cases
    ✓ SignUp page
    ✓ SignUp page content
  Recieve Food- Shelters Page Test Cases
    ✓ shelters page
    ✓ shelters page content
  Settings Page Test Cases
    ✓ Setting page
    ✓ Setting page content
  Negative flow Test Cases
    ✓ Wrong page

  9 passing (124ms)

PS C:\Users\parva\OneDrive\Desktop\sjtu1\spring2020\280\project2020April\FoodStrap\foodstrap>
```

## 02. Selenium Automation

The workflow automation was done using SELENIUM WebDriverJS library for node.js and chrome via chromedriver. The automation script follows the Page Object Model pattern that is often used in test automation. Page Object Model (POM) creates object repository for web UI elements. The motivation in using this pattern was to avoid code redundancy. With the page object model pattern, a class will be defined for each page in the application. The class will contain the UI elements associated with the page and also the actions or operations performed on this page. Using the elements and actions we can write automation scripts.

### 001. Set up

Before developing the automation scripts, the set up required for the successful execution is done. The first step is to install chromedriver . The chromedriver version must match the google chrome version installed in the pc or laptop. There is a chromedriver available for both mac and windows. The version used for the automation of the application is ChromeDriver 81.0.4044.138 and mac64

## Index of /81.0.4044.138/

Name	Last modified	Size	ETag
<a href="#">Parent Directory</a>		-	
<a href="#">chromedriver_linux64.zip</a>	2020-05-05 20:33:58	4.74MB	6581e09a8ce12da2239ac21b2a1cd20b
<a href="#">chromedriver_mac64.zip</a>	2020-05-05 20:34:00	6.70MB	4b2ace862187dc9e53d29c9f12710731
<a href="#">chromedriver_win32.zip</a>	2020-05-05 20:34:01	4.20MB	d19aef5daf9dbaeac152d27066285a7b
<a href="#">notes.txt</a>	2020-05-05 20:34:05	0.00MB	a40835254450bd55cb1c9a1895939357

After the chromedriver installation, the next step is to install the selenium webdriver. The dependency for the same is inside package.json. So, we can just do it by

```
foodstrap (master) $ npm install
```

All the node modules required for the automation script execution will be installed.

## 002. Automation Scripts

Once the setup is complete, the automation scripts are written. As said earlier, page object model pattern was used for building the automation scripts. So before writing the actual automation scripts,

the page classes were defined. One of the page class is illustrated below

```

var webdriver = require('selenium-webdriver');
var BasePage = require('./base');
var By = webdriver.By;
var until = webdriver.until;
var Key = webdriver.Key;
var restdb = class extends BasePage{
    clickOnDonate(){
        driver.wait(until.elementLocated(By.id('donate'))).click();
        return require('./donate')
    }
    donatehistory(){
        driver.wait(until.elementLocated(By.css('#span'))).click();
        driver.wait(until.elementLocated(By.xpath("//a[@href='/donationshistory']"))).click();
        return require('./donhistory')
    }
    donProcess(){
        driver.wait(until.elementLocated(By.css('#span'))).click();
        driver.wait(until.elementLocated(By.xpath("//a[@href='/donationprocess']"))).click();
        return require('./donprocess')
    }
    donProfile(){
        driver.wait(until.elementLocated(By.css('#span'))).click();
        driver.wait(until.elementLocated(By.xpath("//a[@href='/rest_profile']"))).click();
        return require('./restprofile')
    }
    donlogout(){
        driver.wait(until.elementLocated(By.css('#span'))).click();
        driver.wait(until.elementLocated(By.xpath("//a[@href='/']"))).click();
        return require('./home')
    }
}
module.exports = new restdb();

```

The Base Page class defines the constructor for the webdriver as shown below.

```
base.js
var webdriver = require('selenium-webdriver');
var driver = new webdriver.Builder().withCapabilities(webdriver.Capabilities.chrome()).build();

class BasePage{
    constructor() {
        global.driver = driver;
    }
    navigateToHome() {
        driver.get("http://localhost:5000/");
    }
}
module.exports = BasePage;
```

After defining classes for all the pages, the next step is to automate the workflows by building test scripts.

The workflows automated for the application are as follows:

### 1. Restaurant sign up:

Page Class files:

```
signup.js
1 var webdriver = require('selenium-webdriver');
2 var BasePage = require('./base');
3 var By = webdriver.By;
4 var until = webdriver.until;
5 var Key = webdriver.Key;
6 class Signup extends BasePage{
7     fillSignup(){
8         driver.wait(until.elementLocated(By.css('#usertype'))).click();
9         driver.wait(until.elementLocated(By.css('#usertype'))).sendKeys('restaurant');
10        driver.wait(until.elementLocated(By.css('#usertype'))).click();
11        driver.wait(until.elementLocated(By.css('#username'))).sendKeys('rest5');
12        driver.wait(until.elementLocated(By.css('#pass'))).sendKeys('rest5');
13        driver.wait(until.elementLocated(By.css('#rname'))).sendKeys('DinTaiFung');
14        driver.wait(until.elementLocated(By.css('#rcuisine'))).sendKeys('chinese');
15        driver.wait(until.elementLocated(By.css('#rphone'))).sendKeys('1112223666');
16        driver.wait(until.elementLocated(By.css('#remailid'))).sendKeys('dintaifung@gmail.com');
17        driver.wait(until.elementLocated(By.css('#raddr'))).sendKeys('4700 Glenville Rd');
18        driver.wait(until.elementLocated(By.css('#rcity'))).sendKeys('San Ramon');
19        driver.wait(until.elementLocated(By.css('#rstate'))).sendKeys('CA');
20        driver.wait(until.elementLocated(By.css('#rzip'))).sendKeys('94582');
21        driver.wait(until.elementLocated(By.id('signup'))).click();
22        return require('./signin')
23    }
24 }
25 module.exports = new Signup();
```

## Automation file:

```
test.js
1
2 var homepage = require('./home');
3 //var donor= require('./donor');
4 homepage.navigateToHome();
5 var donor = homepage.clickOnDonors();
6 var signup = donor.clickOnSignUp();
7 signup.fillSignup();
```

## 2. Restaurant Manager Making a food donation

## Page Class File:

```
donate.js
1 var webdriver = require('selenium-webdriver');
2 var BasePage = require('./base');
3 var By = webdriver.By;
4 var until = webdriver.until;
5 var Key = webdriver.Key;
6 class donate extends BasePage{
7   confirmdonate(){
8     driver.wait(until.elementLocated(By.css('#menu'))).sendKeys('Spring rolls and fried rice');
9     driver.wait(until.elementLocated(By.css('#cuisine'))).sendKeys('chinese');
10    driver.wait(until.elementLocated(By.css('#count'))).sendKeys('20');
11    driver.wait(until.elementLocated(By.css('#allergy'))).sendKeys('nuts and dairy');
12    driver.wait(until.elementLocated(By.css('#notes'))).sendKeys('pickup curb side');
13    driver.wait(until.elementLocated(By.css('#pickuptime'))).sendKeys('05/06/2020 8:30 PM');
14    driver.wait(until.elementLocated(By.css('#addr'))).sendKeys('4700 Glenville Rd');
15    driver.wait(until.elementLocated(By.css('#city'))).sendKeys('San Ramon');
16    driver.wait(until.elementLocated(By.css('#state'))).sendKeys('CA');
17    driver.wait(until.elementLocated(By.css('#zip'))).sendKeys('94582');
18    driver.wait(until.elementLocated(By.css('#confirmdonate'))).click();
19    console.log("after");
20  }
21  module.exports = new donate();
```

## Automation File:

```
testdonate.js
1 var homepage = require('./home');
2 homepage.navigateToHome();
3 var signin = homepage.clickOnSignin();
4 var restdb = signin.signinrest();
5 var donate = restdb.clickOnDonate();
6 donate.confirmdonate();
```

## 3. Shelter Manager claiming the food donation

## Page Class file:

```
claim.js
1 var webdriver = require('selenium-webdriver');
2 var BasePage = require('./base');
3 var By = webdriver.By;
4 var until = webdriver.until;
5 var Key = webdriver.Key;
6 class claim extends BasePage{
7     confirmclaim(){
8
9         driver.wait(until.elementLocated(By.xpath("//a[@href='/claimdonation']"))).click();
10        return require('./shelterdb');
11    }
12}
13 module.exports = new claim();
```

Automation file:

```
testshelterclaim.js
1 var homepage = require('./home');
2 homepage.navigateToHome();
3 var signin = homepage.clickOnSignin();
4 var shelterdb = signin.signinshelt();
5 var claim = shelterdb.claim();
6 var confclaim = claim.claimdon();
7 confclaim.confirmclaim();
```

#### 4. Volunteer associated with the shelter picking up the food donation

Page Class File:

```
pickup.js
1 var webdriver = require('selenium-webdriver');
2 var BasePage = require('./base');
3 var By = webdriver.By;
4 var until = webdriver.until;
5 var Key = webdriver.Key;
6 class pickup extends BasePage{
7     confirmpickup(){
8
9         driver.wait(until.elementLocated(By.css("#confirmpickup"))).click();
10        return require('./voldb');
11    }
12}
13 module.exports = new pickup();
```

Automation file:

```
testvolpickup.js
1 var homepage = require('./home');
2 homepage.navigateToHome();
3 var signin = homepage.clickOnSignin();
4 var voldb = signin.signinvol();
5 var pickup = voldb.pickup();
6 var confpickup = pickup.confirmpickup();
7
```

## 5. Navigation to donation process page in restaurant dashboard

Page Class File:

```
restdb.js
1 var BasePage = require('./base');
2 var By = webdriver.By;
3 var until = webdriver.until;
4 var Key = webdriver.Key;
5 class restdb extends BasePage{
6     clickOnDonate(){
7         driver.wait(until.elementLocated(By.id('donate'))).click();
8         return require('./donate')
9     }
10    donatehistory(){
11        driver.wait(until.elementLocated(By.css('#span'))).click();
12        driver.wait(until.elementLocated(By.xpath("//a[@href='/donationshistory']"))).click();
13        return require('./donhistory')
14    }
15    donProcess(){
16        driver.wait(until.elementLocated(By.css('#span'))).click();
17        driver.wait(until.elementLocated(By.xpath("//a[@href='/donationprocess']"))).click();
18        return require('./donprocess')
19    }
20    donProfile(){
21        driver.wait(until.elementLocated(By.css('#span'))).click();
22        driver.wait(until.elementLocated(By.xpath("//a[@href='/rest_profile']"))).click();
23        return require('./restprofile')
24    }
25    donlogout(){
26        driver.wait(until.elementLocated(By.css('#span'))).click();
27        driver.wait(until.elementLocated(By.xpath("//a[@href='/']"))).click();
28        return require('./home')
29    }
30}
```

Automation File:

```
testrestprocess.js
1 var homepage = require('./home');
2 homepage.navigateToHome();
3 var signin = homepage.clickOnSignin();
4 var restdb = signin.signinrest();
5 var dprocess = restdb.donProcess();
```

## 6. Shelter History page in shelter dashboard

Page Class File:

```

shelterdb.js
[  var webdriver = require('selenium-webdriver');
  var BasePage = require('./base');
  var By = webdriver.By;
  var until = webdriver.until;
  var Key = webdriver.Key;
  class shelterdb extends BasePage{
    claim(){
      driver.wait(until.elementLocated(By.css('#span'))).click();
      driver.wait(until.elementLocated(By.xpath("//a[@ref='/shelter_claim']"))).click();
      return require('../sheltclaim')
    }
    shelthistory(){
      driver.wait(until.elementLocated(By.css('#span'))).click();
      driver.wait(until.elementLocated(By.xpath("//a[@ref='/past_claim']"))).click();
      return require('../shelthistory')
    }
    sheltProfile(){
      driver.wait(until.elementLocated(By.css('#span'))).click();
      driver.wait(until.elementLocated(By.xpath("//a[@ref='/shelt_profile']"))).click();
      return require('../sheltpfile')
    }
    shelLogout(){
      driver.wait(until.elementLocated(By.css('#span'))).click();
      driver.wait(until.elementLocated(By.xpath("//a[@ref='/']"))).click();
      return require('../home')
    }
  }
]

```

### Automation File

```

testshelthistory.js
[  var homepage = require('../home');
  homepage.navigateToHome();
  var signin = homepage.clickOnSignin();
  var shelterdb = signin.signinshelt();
  var shistory = shelterdb.shelthistory();
]

```

### 7. Updating volunteer profile page in volunteer dashboard

#### Page Class File

```

voldb.js
1 var webdriver = require('selenium-webdriver');
2 var BasePage = require('./base');
3 var By = webdriver.By;
4 var until = webdriver.until;
5 var Key = webdriver.Key;
6 class voldb extends BasePage{
7     pickup(){
8
9         driver.wait(until.elementLocated(By.css('#pickup'))).click();
0         return require('./pickup')
1     }
2
3     volProfile(){
4         driver.wait(until.elementLocated(By.css('#span'))).click();
5         driver.wait(until.elementLocated(By.xpath("//a[@href='/vol_profile']"))).click();
6         return require('./volprofile')
7     }
8     volLogout(){
9         driver.wait(until.elementLocated(By.css('#span'))).click();
0         driver.wait(until.elementLocated(By.xpath("//a[@href='/']"))).click();
1         return require('./home')
2     }
3 }
4 module.exports = new voldb();
5

```

Automation File

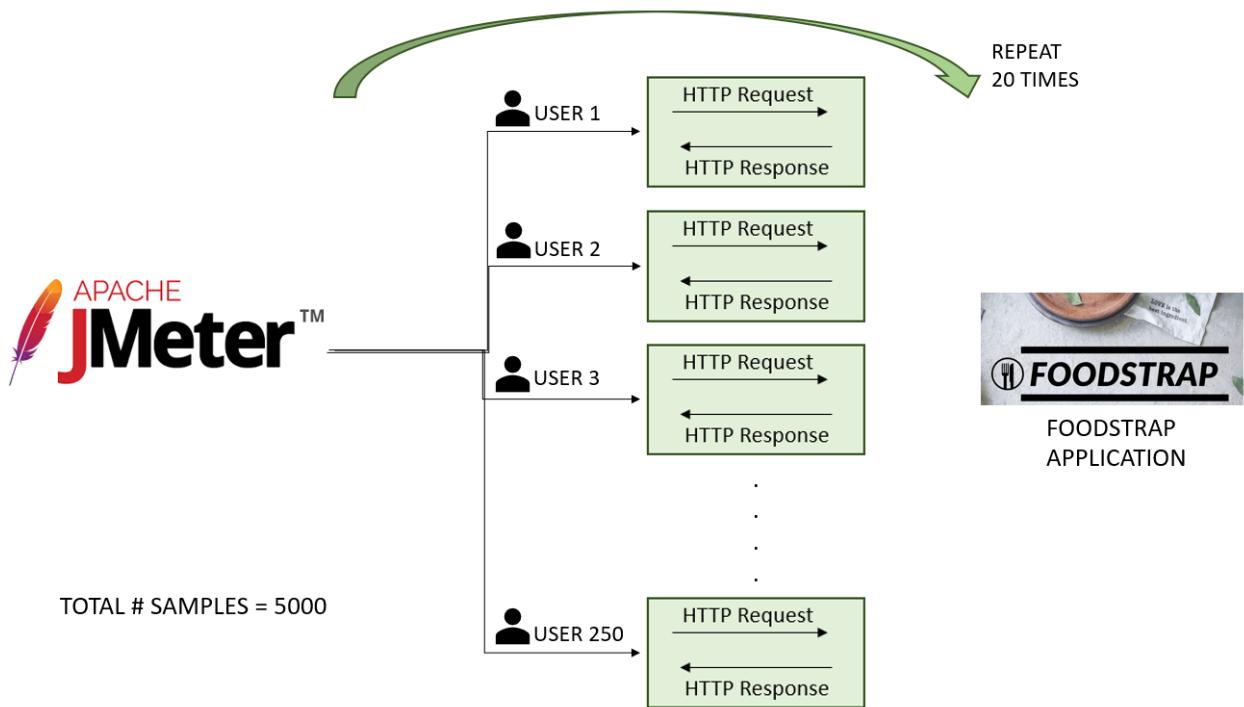
```

testvolprofile.js
1 var homepage = require('./home');
2 homepage.navigateToHome();
3 var signin = homepage.clickOnSignin();
4 var voldb = signin.signinvol();
5 var profile = voldb.volProfile();
6 profile.updateprofile();

```

## 16. Load Testing using JMeter

Apache JMeter, a Java open source software for performance testing, was used for testing the application. Total of 5000 requests were sent to the server. To simulate multiple user requests to the application, the number of threads is given as 250. JMeter will create and simulate 250 user requests. Loop count is given as 20, so this process is repeated 20 times for a total of 5000 requests to the application. The listeners are used to display the result of the test execution. Formats used are table, tree, summary and aggregate reports, graph and response time graph.



Thread Group showing number of threads and loop count:

File Edit Search Run Options Tools Help

FoodStrapTest Plan.jmx (C:\Users\parva\OneDrive\Desktop\sjsu1\spring2020\280\project2020April\apache-jmeter-5.3\bin\FoodStrapTest Plan.jmx) - Apache JMeter (5.3)

FoodStrapTest Plan

Users

- IndexPage
- View Results in Table
- View Results Tree
- Summary Report
- Graph Results
- Aggregate Report
- Response Time Graph
- Aggregate Graph

Thread Group

Name:

Comments:

Action to be taken after a Sampler error:
  Continue
  Start Next Thread Loop
  Stop Thread
  Stop Test
  Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count:
  Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Summary report

**Summary Report**

Name:	Summary Report			
Comments:				
- Write results to file / Read from file				
Filename				
Label	# Samples	Average	Min	Max
IndexPage	5000	1248	16	20519
TOTAL	5000	1248	16	20519

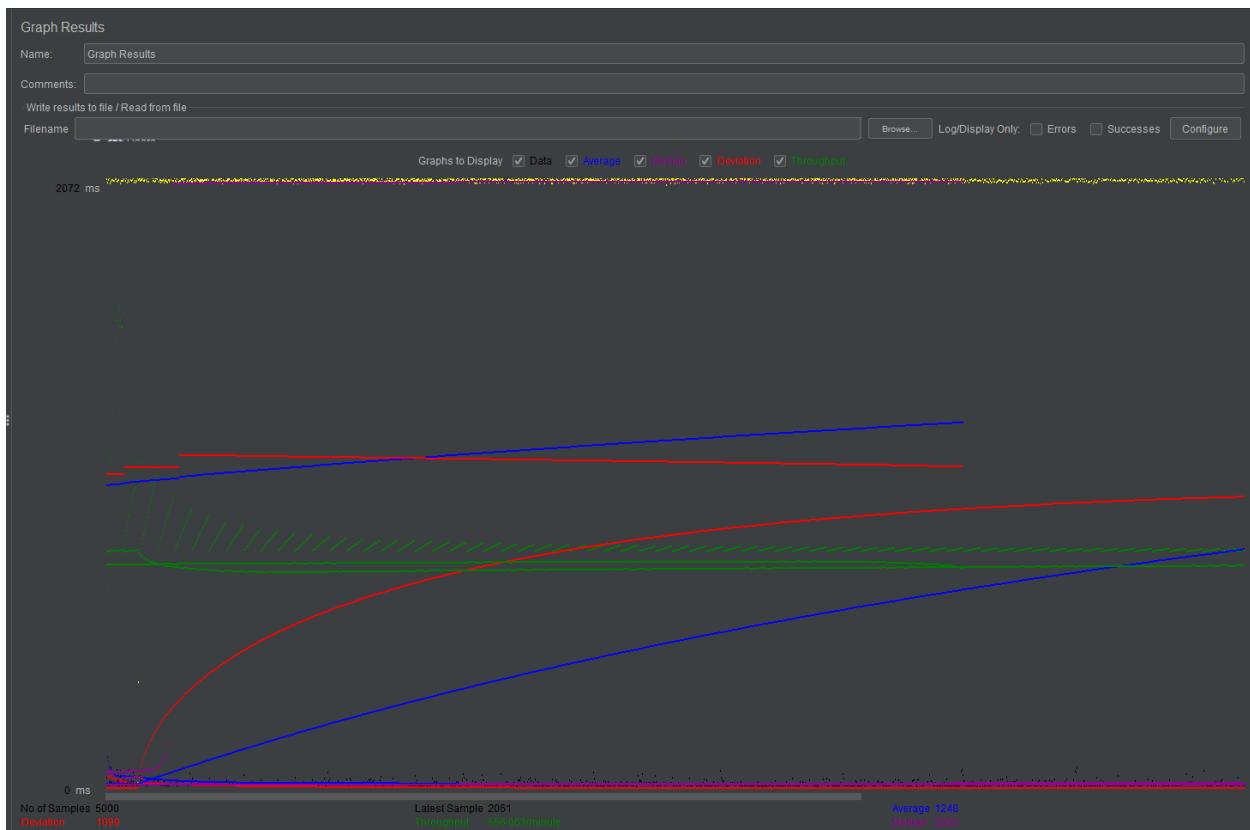
View Results in Table Listener showing the test results:

View Results in Table

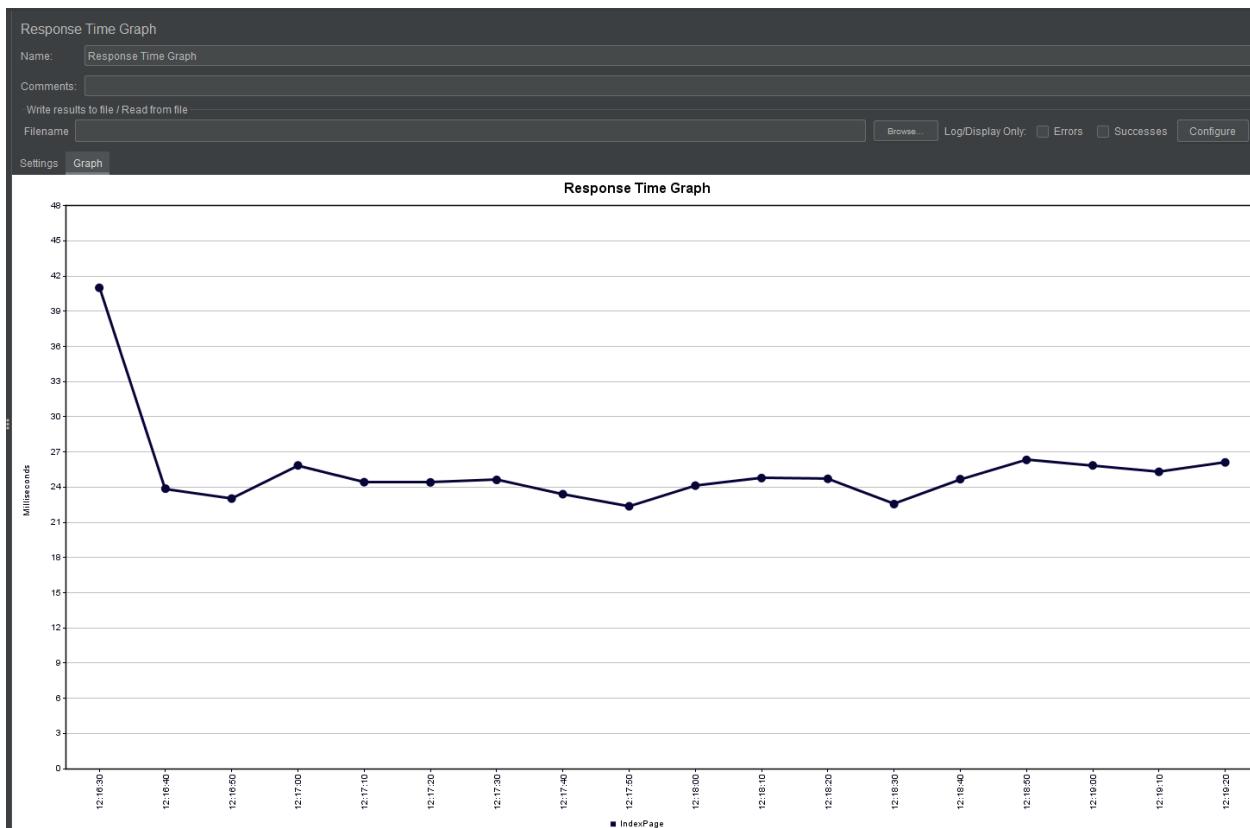
Name:	View Results in Table								
Comments:									
Write results to file / Read from file									
Filename									
		Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="checkbox"/> Configure			
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(m..)
1	12:16:35.667	Users 1-1	PageIndex	165	✓	9540	157	165	45
2	12:16:35.832	Users 1-1	PageIndex	68	✓	9540	157	68	0
3	12:16:35.900	Users 1-1	PageIndex	70	✓	9540	157	70	0
4	12:16:35.970	Users 1-1	PageIndex	27	✓	9540	157	27	0
5	12:16:35.998	Users 1-1	PageIndex	28	✓	9540	157	27	0
6	12:16:36.026	Users 1-1	PageIndex	80	✓	9540	157	80	0
7	12:16:36.106	Users 1-1	PageIndex	23	✓	9540	157	23	0
8	12:16:36.129	Users 1-1	PageIndex	46	✓	9540	157	46	0
9	12:16:36.175	Users 1-1	PageIndex	26	✓	9540	157	26	0
10	12:16:36.202	Users 1-1	PageIndex	26	✓	9540	157	26	0
11	12:16:36.228	Users 1-1	PageIndex	24	✓	9540	157	24	0
12	12:16:36.252	Users 1-1	PageIndex	25	✓	9540	157	25	0
13	12:16:36.277	Users 1-1	PageIndex	72	✓	9540	157	72	0
14	12:16:36.350	Users 1-1	PageIndex	30	✓	9540	157	29	0
15	12:16:36.380	Users 1-1	PageIndex	60	✓	9540	157	60	0
16	12:16:36.440	Users 1-1	PageIndex	23	✓	9540	157	23	0
17	12:16:36.464	Users 1-1	PageIndex	67	✓	9540	157	67	0
18	12:16:36.531	Users 1-1	PageIndex	65	✓	9540	157	65	0
19	12:16:36.596	Users 1-1	PageIndex	51	✓	9540	157	51	0
20	12:16:36.647	Users 1-1	PageIndex	22	✓	9540	157	22	0
21	12:16:37.672	Users 1-2	PageIndex	66	✓	9540	157	66	13
22	12:16:37.740	Users 1-2	PageIndex	65	✓	9540	157	65	0
23	12:16:37.806	Users 1-2	PageIndex	21	✓	9540	157	21	0
24	12:16:37.827	Users 1-2	PageIndex	48	✓	9540	157	48	0
25	12:16:37.875	Users 1-2	PageIndex	21	✓	9540	157	21	0

Scroll automatically?  Child samples? No of Samples 5000 Latest Sample 2061 Average 1248 Deviation 1099

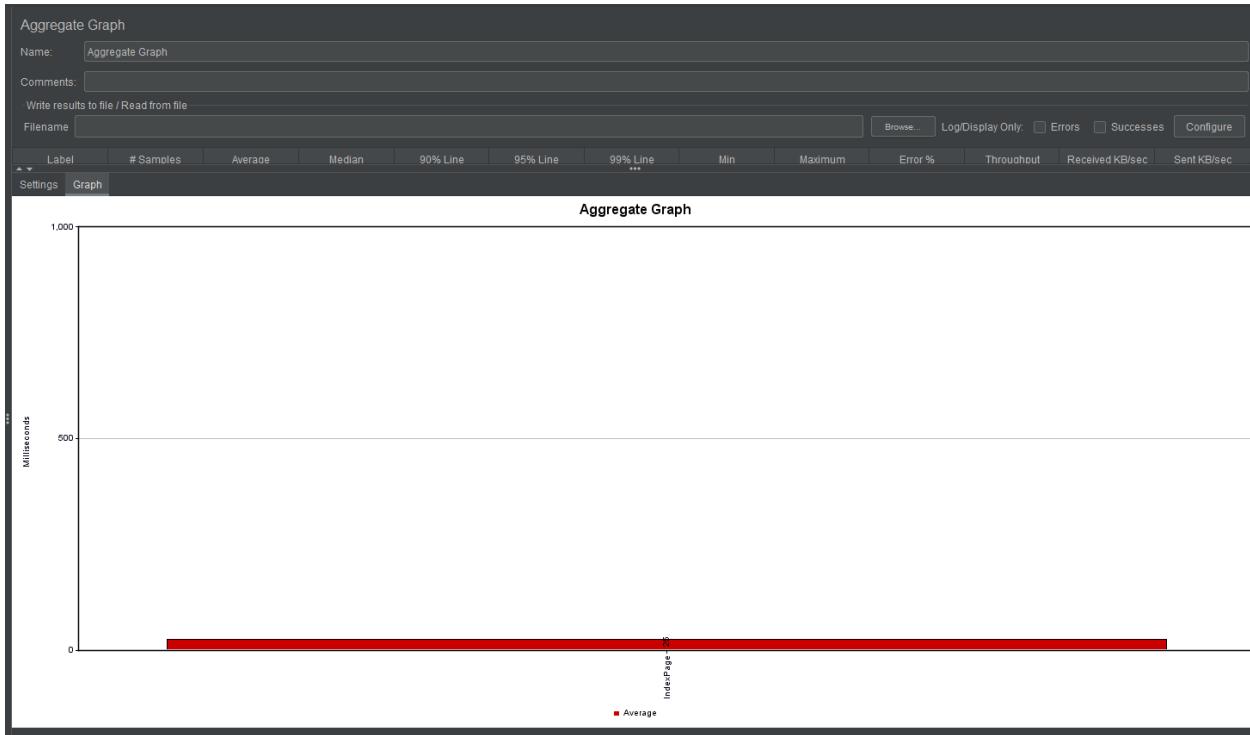
Graph results Listener showing the test results:



Response time Graph Listener showing the test results:



Aggregate Graph Listener showing the test results:



## 17. Cross Browser Compatibility

Cross Browser Compatibility is the ability of a website or web application to function across different browsers and degrade gracefully when browser features are absent or lacking. Also, the web application has been achieved using Bootstrap library. Bootstrap supports all major browsers and also supports browsers in android mobile devices and iPhone.

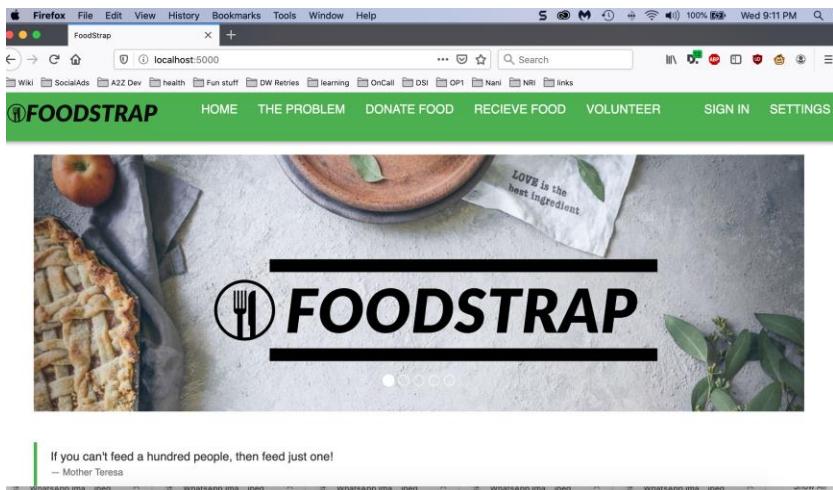
Code Snapshot of bootstrap file included in the ejjs files:

```
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="css/custom.css">
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
```

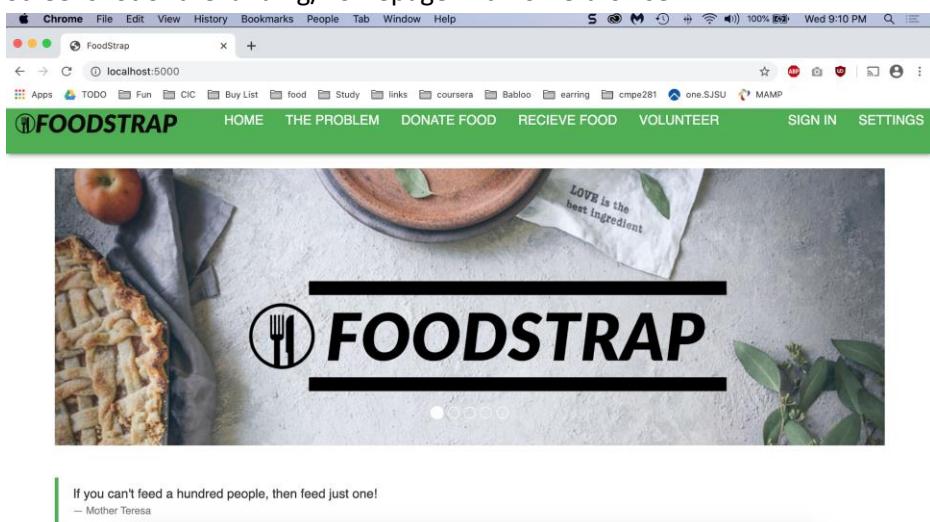
For Internet Explorer, we have included

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

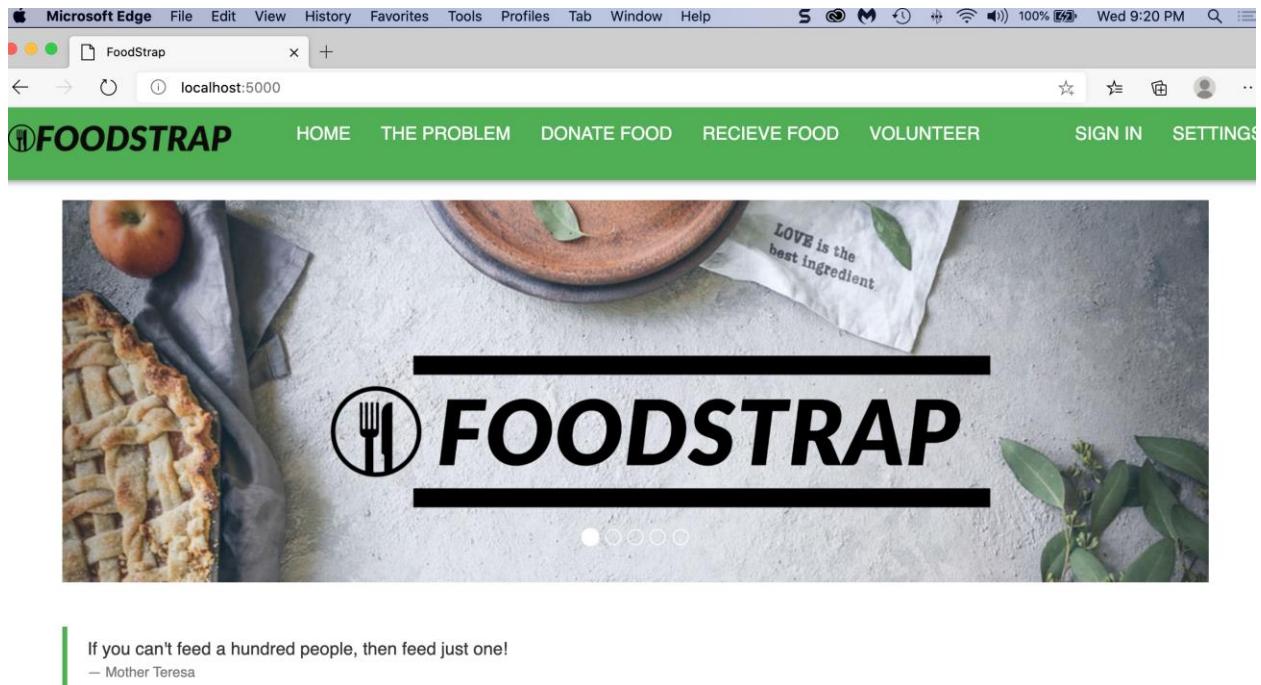
Screenshot of the landing/homepage in Firefox browser:



Screenshot of the landing/homepage in chrome browser:



Screenshot of the landing/homepage in Edge browser:



If you can't feed a hundred people, then feed just one!  
— Mother Teresa

## 01. Cross Device Compatibility

The cross-device compatibility has been achieved via bootstrap and also by setting the viewport through meta tag.

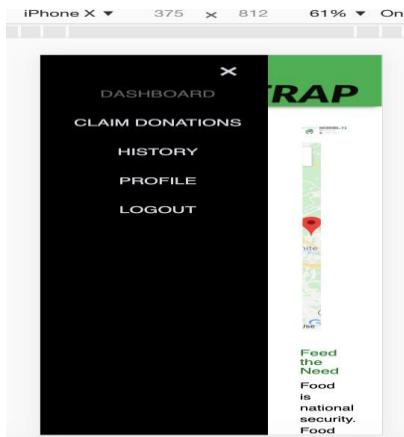
```
<!DOCTYPE html>
<html lang="en">

<head>
  <title><%= msgs.title %></title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
```

We have also achieved responsiveness for certain elements like the responsive side navbar by adding media screen properties to them :

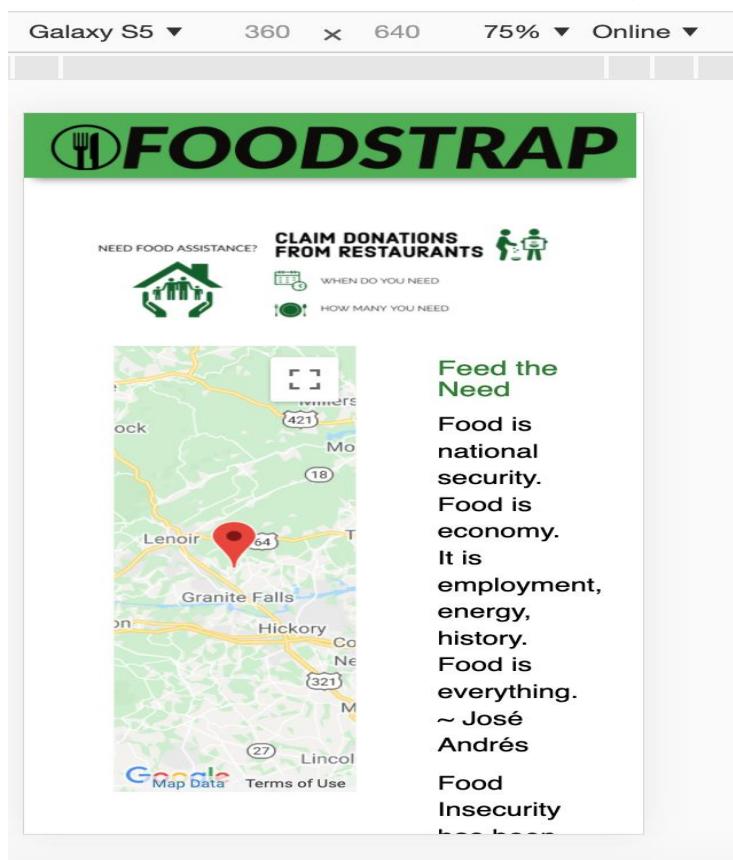
```
/* On smaller screens, where height is less than 450px, change the style of the sidenav (less padding and
@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
})
```

Screenshot for the sidenav in mobile device:

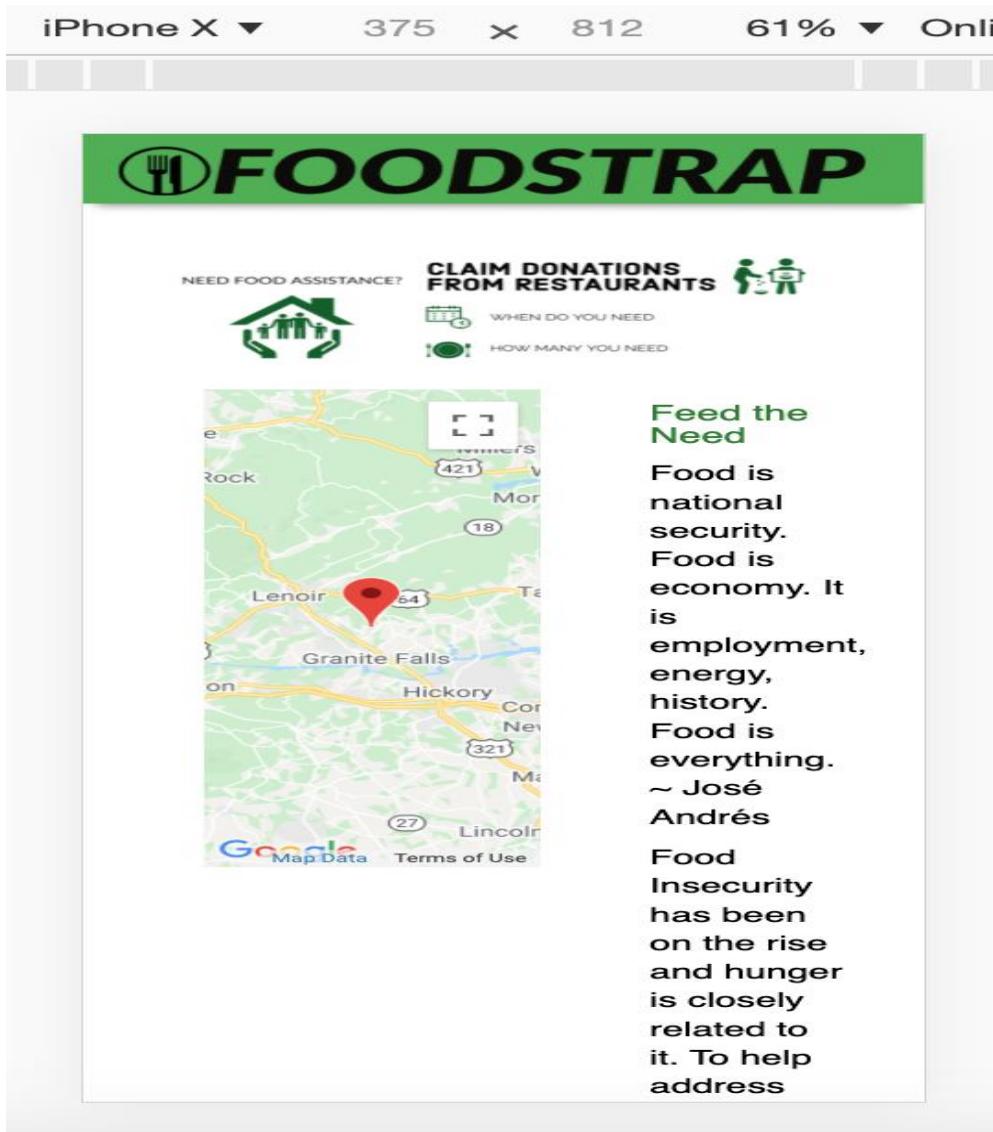


Screenshot of shelter dashboard on iPad pro:

Screenshot of shelter dashboard on mobile (android):



Screenshot of shelter dashboard on iPhone:



## 18. Library

### 01. Front end JavaScript Library

#### jQuery

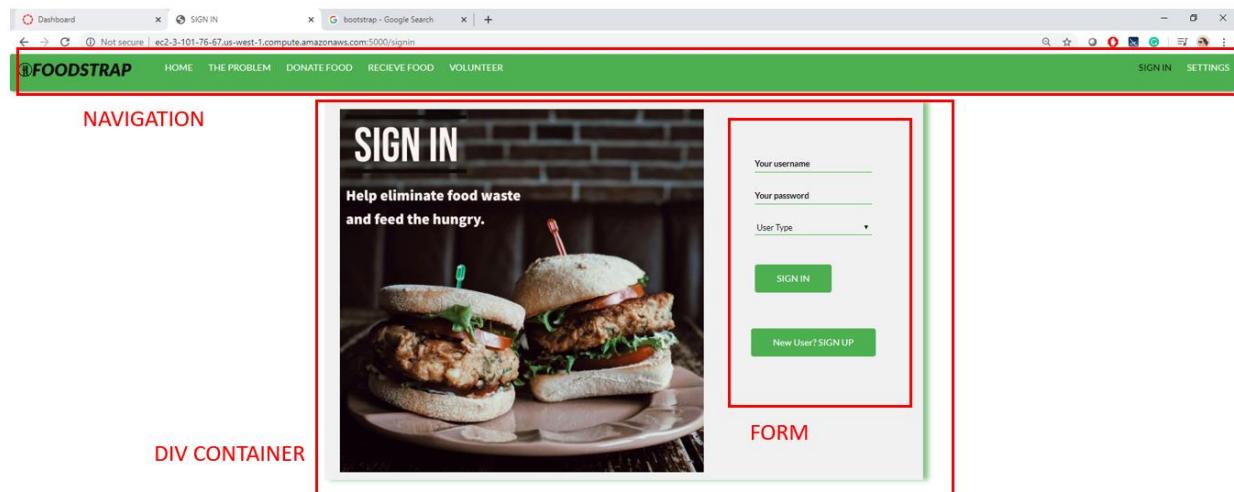
jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation. It is used as it simplifies coding with few lines of code. For instance, data tables are displayed using jQuery. Sample code used is:

```
$(document).ready(function(){
    $('#myTable').dataTable();
});
```

## Bootstrap

Bootstrap contains CSS and JavaScript based design templates for typography, forms, buttons, navigation, and other interface components. In the application, bootstrap templates used are for form, button, div classes like container and top navigation. Sample code for the same is:

- <div class="container greyShadow">
- <nav class="navbar navbar-default">
- <div class="form-group form-input">



## Bootstrap-datetimepicker

A date/time picker component designed to work with Bootstrap. The date time picker is used in donations form to enter pick up date and time for a food donation.

The image contains two screenshots of a web-based donation form. Both screenshots show the following fields:

- Pick Up Time:** A text input field containing "05/19/2020 12:00 AM".
- Pick Up Address:** A text input field containing "934 N1st Street".
- City:** A text input field containing "San Jose".
- Zip:** A text input field containing "95112".
- Donate:** A green button.

The top screenshot shows a date picker modal for "May 2020". The date "19" is selected. The bottom screenshot shows a time picker modal with hours set to 12, minutes to 00, and AM selected.

### Moment.js

This used to display and format date time along with the date picker library in donations form.

### D3js

D3.js is a JavaScript library for producing dynamic, interactive data visualizations. It is used for creating and displaying a word cloud in the Problems page. As a word cloud shows the important keywords, it is better than a paragraph of text to capture user attention.



# THE PROBLEM

## Food Insecurity

poverty affordable problem  
wages lack  
**insecurity**  
us needs financial  
**resources** people **Food**

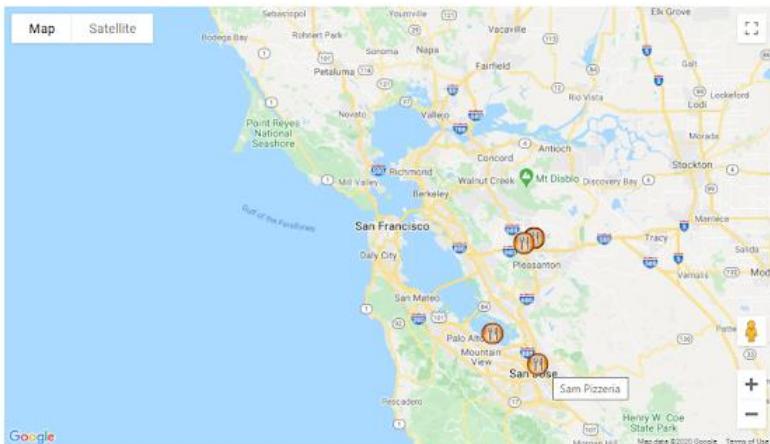
## Google Maps

Google maps with markers and are added to pages to show the required locations. The donation location is shown on a map in shelters' claim and volunteers pick up page. All the restaurants who make donations are shown on the map in the donors page. The shelter location is also shown on a map in the dashboard. This is dynamically populated for each donation, restaurant and shelter.

The simplest acts of kindness are by far more powerful than a thousand heads bowing in prayer.  
— Mahatma Gandhi

Many families in our area need a little help putting food on the table. They shouldn't have to choose between paying rent or buying healthy groceries. You can help by making food donations. Your support is greatly needed and appreciated. In the US, food waste is estimated at between 30-40% of our food supply, estimated at around 72 billion-133 billion pounds of wasted food each year, according to the USDA. Rather than toss unsold breads, pastries, and produce to a dumpster, box them up for local nonprofits and those in need.

From the Michelin-starred, national restaurant chains to world-wide mega coffee shops that proliferate every street corner in America, check out these establishments that are committed to cut down on food waste by donating at the end of the day.



If you have food you'd like to donate to a local shelter, sign up now!

[SIGN UP NOW!](#)

**Menu**

**Fried Pasta**

---

Cuisine	Portion Count
Italian	30

---

Allergy Information

Nuts, diary

---

Special Instructions

None

---

Pick Up Time

04/30/2020 10:00 PM

---

Pick Up Address

935 N2nd Street, San Jose, CA 95112

---

Status

OPEN

[CANCEL](#) [CONFIRM CLAIM](#)

## Google Charts

### Geo chart

A *geochart* is a map of a country, a continent, or a region with areas identified and styles. The regions style fills entire regions (typically countries) with colors corresponding to the values that you assign. It is used to show the number of donations made by each country in the index page and by each restaurant in the restaurant dashboard. This is dynamically populated.

#### Share Your Food

Many families in our area need a little help putting food on the table. They shouldn't have to choose between paying rent or buying healthy groceries. You can help by making food donations. Your support is greatly needed and appreciated.

#### Donate Surplus Food

Instead of throwing away surplus food that is safe to eat, donate to local hunger relief organizations. This delivers a triple benefit: Food stays out of landfills to prevent environmental impact, nonprofits benefit from the donations and restaurants give back to local communities.

#### Receive Food

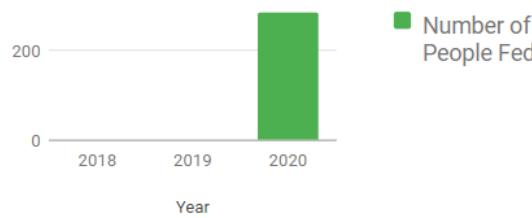
The quality, wholesome prepared food donated from restaurants can serve those who experience hunger. If you have to choose between paying rent or buying healthy groceries, we can help.



### Bar charts

Bar charts are used in some pages to visually depict the parameters and its values. It is used in donations history and claims history pages. This is dynamically populated for each user after collecting the value from the database.

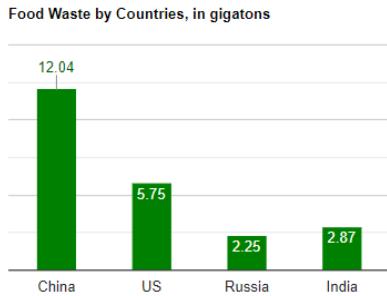
Donations Summary



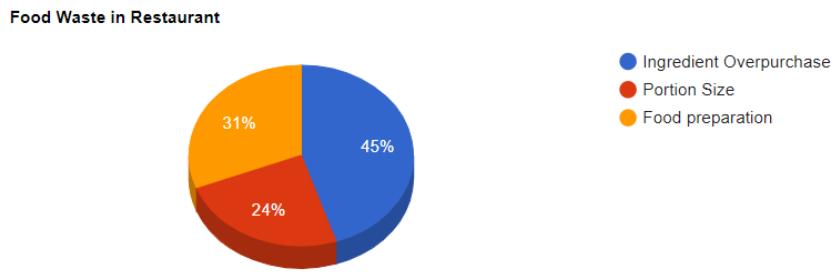
### Other charts

Column and Pie charts are used to depict the food waste parameters in the Problems page. These are static charts.

## Food Waste.



Up to 40 percent of the food in the United States is never eaten. Of the estimated 125 to 160 billion pounds of food that goes to waste every year, much of it is perfectly edible and nutritious. Food is lost or wasted sometimes because of overproduction.



## 02. NodeJS Library

### 1. Express

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It makes creating APIs easy and provides robust routing. It is designed for building web applications and APIs.

```

2  var express = require('express');
3  var path = require('path');
4  var cookieParser = require('cookie-parser');
5  var logger = require('morgan');
6  var session = require('client-sessions');
7  var router = require('./routes/route');
8  var app = express();      parvathysjsu [7 weeks ago] • initia
9
10 // view engine setup
11 app.set('views', path.join(__dirname, 'views'));
12 app.set('view engine', 'ejs');
13 > app.use(session({ ...
14 });
15 app.use(logger('dev'));
16 app.use(express.json());
17 app.use(express.urlencoded({ extended: false }));
18 app.use(cookieParser());
19 app.use(express.static(path.join(__dirname, 'public')));
20
21 app.use('/', router);

```

```

11  var express = require('express');
12  var router = express.Router();
13
14
15 > router.get('/error_msg', function (req, res, next) { ...
16  });
17  /* SHOW landing PAGE */
18 > router.get('/', function (req, res, next) { ...
19  });
20
21
22 > router.get('/restList', function (req, res, next) { ...
23  });
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

## 2. ejs

EJS is embedded JavaScript templates. It is used as the view engine for the Express application. It is a simple templating language that lets you generate HTML markup with plain JavaScript. This is used for all pages in the application.

```

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');      parvathysjsu [7 weeks ago] • initia

```

The screenshot shows a code editor with the following file structure:

```

index.ejs
  |
  +-- views
      +-- index.ejs
          +-- html
              +-- body

```

The content of index.ejs is:

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <title><%= msgs.title %></title>
6      <meta name="FoodStrap" content="FoodStrap Making it easy to donate food.">

```

### 3. Client-sessions

client-sessions is connect middleware that implements sessions in encrypted tamper-free cookies. This is used to manage session data on the client side. We specify a cookie name that dictates the key name added to the request object and duration for how long the session will stay valid in ms

```

6  var session = require('client-sessions');
7  var router = require('./routes/route');
8  var app = express();
9
10 // view engine setup
11 app.set('views', path.join(__dirname, 'views'));
12 app.set('view engine', 'ejs');
13 app.use(session({
14     cookieName: 'session',
15     secret: 'easywaytodonatefood',
16     duration: 30 * 60 * 1000,
17     activeDuration: 5 * 60 * 1000,
18 }));

```

### 4. Cookie-parser

Parse Cookie header and populate req.cookies with an object keyed by the cookie names.

```

6  var cookieParser = require('cookie-parser');
7
8  app.use(cookieParser());
9

```

### 5. Http-errors

It used to create HTTP errors for Express APIs.

```

var createError = require('http-errors');

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

```

## 6. Mongodb

It is the MongoDB driver for Node.js and provides a high-level API. It is used for performing CRUD operations on collections in MongoDB. MongoClient is used to connect to mongoDB and perform CRUD operations on collections. ObjectId is used for creating new mongodb identifier \_id object.

```

MongoClient.connect("mongodb://localhost:27017/foodstrap", function (err, db) {
  if (!err) {
    console.log("We are connected");
  }
  var dbo = db.db("foodstrap");
  var myquery = { _id: new ObjectId(donid) };
  var newvalues = { $set: don };
  dbo.collection("donations").updateOne(myquery, newvalues, function (err, resp) {
    if (err) throw err;
    console.log("1 document updated");
    res.redirect("/volunteer_dashboard");
  });
});
});
```

## 7. Morgan

Morgan is the HTTP request logger middleware for node.js. It allows us to easily log requests, errors to console.

```

var logger = require('morgan');

app.use(logger('dev'));

```

## 8. Node-geocoder

Node-geocoder is the geocoder API in NodeJS. Geocoding is the process of taking input text, such as an address and returning a latitude/longitude location on the Earth's surface for that place. When users are giving their address, they are not aware of the location's latitude and longitude. Latitude and longitude are required to display the location on google maps. Using the geocoder API, we fetch the latitude and

longitude of an address. It is used for restaurant addresses when a restaurant signs up in the app, food donation pick up location when a restaurant manager makes a donation. Using the results of this API, the google maps are displayed.

```
var geoCoder = NodeGeocoder({
  provider: 'openstreetmap'
});
console.log("---addr searched: " + addr);
geoCoder.geocode(addr)
  .then((mapres) => {
    console.log(mapres);
    let position = {};
    let lat = mapres[0].geometry.location.lat;
    let lon = mapres[0].geometry.location.lng;
    position['lat'] = lat;
    position['lon'] = lon;
    return position;
  })
  .catch((err) => {
    console.log(err);
  });
}

module.exports = geoCoder;
```

### 03. Library used in Test Automation

- Mocha

Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser for asynchronous testing. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. Mocha is used for test automation for the application.

Command to run mocha test cases: npm test

Script Location: FoodStrap\foodstrap\test

```
foodstrap > {} package.json > {} scripts > test
1  {
2    "name": "foodstrap",
3    "version": "0.0.1",
4    "private": true,
5    "scripts": [
6      "start": "node app.js",
7      "test": "./node_modules/.bin/mocha"
8    ],
9  }
```

- Chai

Chai is an assertion library for node. When running test cases using mocha, we use chai to assert the results. It provides several assertion methods like assert, expect and should. Using this we can easily check the results of test case against expected result.

```
foodstrap > test > JS testcases.js > [e] expect
1 var expect = require('chai').expect;
2
3
4 function verifyPageError(response) {
5   expect(response.statusCode).to.equal(404);
6 }
7 function verifyPage(response) {
8   expect(response.statusCode).to.equal(200);
9 }
```

- Cheerio

Cheerio is the fast, flexible & lean implementation of core jQuery designed specifically for the server. Cheerio parses markup and provides an API for traversing/manipulating the resulting data structure. It does not interpret the result as a web browser does. When running mocha test cases, we get html code as output. To parse them cheerio is used in place of jQuery. Cheerio implements a subset of core jQuery.

```
function verifyPageContent(body) {
  var $ = cheerio.load(body);
  var submit = $('#submit').attr('name');
  Promise.resolve(submit)
    .then((submit) => {
      //console.log(submit);
      expect(submit).to.equal('submit');
    })
}
```

- Request

Request is designed to be the simplest way possible to make http calls. It supports HTTPS and follows redirects by default. In mocha test cases, it is used to make http calls to the application and get responses. This response is then verified.

```

var request = require('request');      parvathysjsu [3 weeks ago] • mocha tc
var cheerio = require('cheerio');

Run Test | Debug Test
✓ describe('Automated test Cases', function () {
  Run Test | Debug Test
    ✓ describe('SignIn Page Test Cases', function () {
      Run Test | Debug Test
        it('SignIn page', function () {
          request('http://localhost:5000/signin', function (error, response, body) {
            verifyPage(response);
          });
        });
    });
}

```

- selenium-webdriver

Selenium is a browser automation library, used for testing web-applications. It is used for writing automated selenium test cases for various flows. Selenium may be used for any task that requires automating interaction with the browser. We also use chromedriver along with it.

```

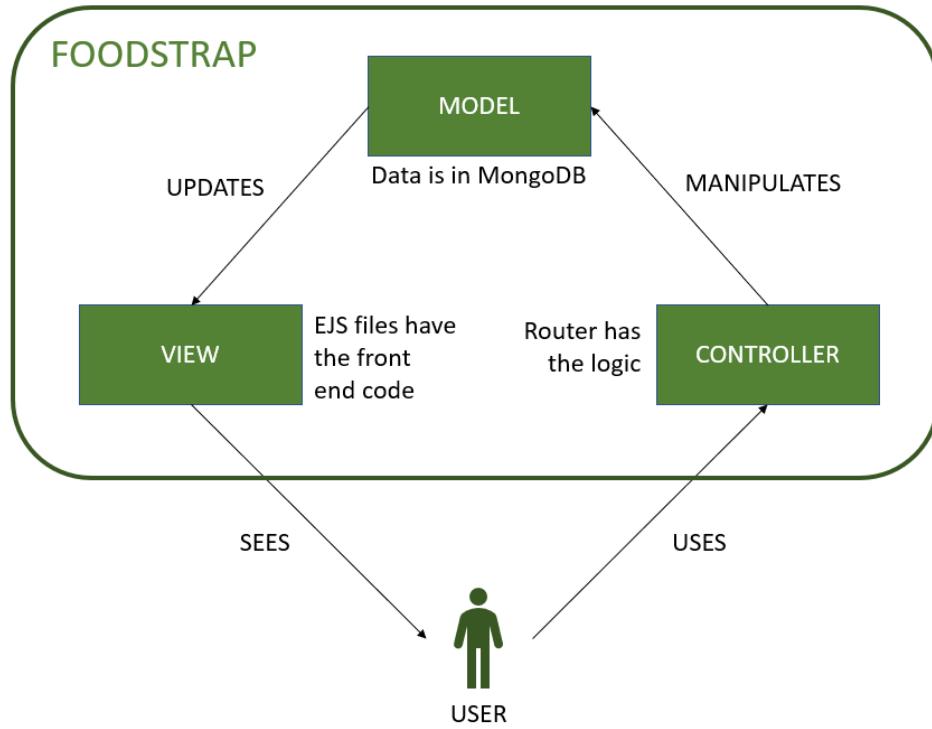
Selenium > JS base.js > ...
1 var webdriver = require('selenium-webdriver');
2 var driver = new webdriver.Builder().withCapabilities(webdriver.Capabilities.chrome()).build();
3
4 class BasePage{
5   constructor() {
6     global.driver = driver;
7   }
8 }

```

## 19. Design Patterns

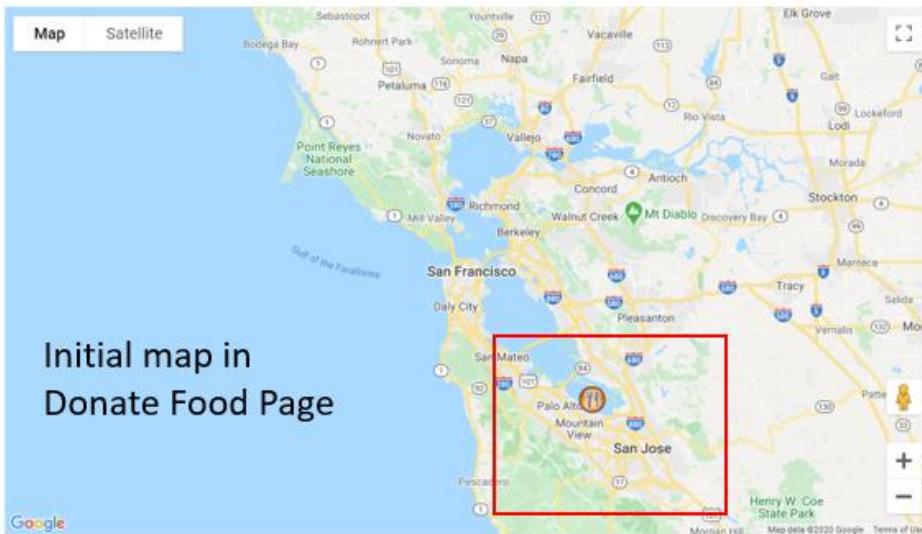
### MVC

The Model View Controller (MVC) design pattern is used. The application is designed in such a way that there is a separate model layer, view layer and controller layer. The front-end code is in views/ folder. For each page, there is a separate ejs file which has the code for that file. All the back-end logic for the controller is in the router file. The data is stored in MongoDB and the router handles all calls to the database.



### Observer Design Pattern

When one object or data is modified, some other objects which depend on this object for data should also change. In the application, there are some elements which depend on other objects. Observer pattern is used for this. It notifies the dependent observers of any change caused to the subject object. All the maps and graphs used in the application are dependent on the data in the database. When a new donation is made or an existing donation is claimed or picked up, the charts need to be updated. When a new restaurant signs up in the app, the maps should be updated to show the new restaurant also. For example, there is a map in the Donate food page. It shows all the restaurants signed up in the app. When a new restaurant signs up in the app, this map is updated to show the new restaurant as well.



Initial map in  
Donate Food Page



Map is updated after  
A new restaurant  
signed up

### jQuery Narrowing Selection

To filter the elements required, narrowing the selection is used. The filter() used, will narrow the objects and provide the required elements only.

### jQuery Method Chaining

To apply multiple methods on the same objects, method chaining is used.

## 20. Pagination

When displaying a large number of records, pagination is required. In the application, pagination is used in the following pages

- Restaurant's Donations History page
- Shelter's Past Claims page
- Shelter's Donation Claim page
- Volunteer's Dashboard page for Pick up Donations

#### Example: Restaurant's Donations History page

In this page, all the donations made by the restaurant are shown. Since the number of records are high, pagination is required. This allows the user to see a limited number of records at once and can use pagination to go to the next set of records.

FoodStrap Project Report										
Show <input type="button" value="10"/> entries <input type="text" value="Search:"/>										
#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
1	taco	mex	20	04/14/2020 12:00 AM	ffg, 2, 2 94020	nuts	come	Home For Good	Shawn Booth	CLAIMED
2	burrito	mexican	10	04/14/2020 8:00 PM	sdf, sf, ca 95112	nuts, peanut	come and pick up at 8			OPEN
3	Salsa and Chips	mexican	30	04/20/2020 8:00 PM	912 N2nd Street, San Jose, CA 95110	Nuts, diary	come and pick up at 8		Shawn Booth	DONE
4	salsa and tacos	Mexican	50	04/29/2020 7:00 PM	935 N2nd Street, San Jose, CA 95112	nuts	abc	Home For Good	Shawn Booth	DONE
5	burgers	American	25	04/29/2020 8:00 PM	922 N 2nd street, San Jose, CA 95112	Nuts, diary	Curb side pick up	Home For Good	Shawn Booth	DONE
6	Salsa and Chips	Mexican	25	05/04/2020 11:00 PM	Moffet Field, Mountain View, CA 94035	Nuts, diary	None			OPEN
7	Veg Pasta	Italian	22	05/03/2020 11:00 PM	Moffet Field, Mountain View, CA 94035	Diary	Curb side pick up			OPEN
8	Pizza and Garlic Bread	Pizza	20	04/14/2020 12:00 AM	937 N2nd Street, San Jose, California 95112	Nuts, diary	Curb side pick up			OPEN
9	taco	Mexican	10	05/05/2020 12:00 AM	Moffet Field, Mountain View, CA 94035	Nuts, diary	None			OPEN
10	Veg Pasta	Italian	30	04/28/2020 12:00 AM	Moffet Field, Mountain View, CA 94035	Diary	None			OPEN

Showing 1 to 10 of 11 entries

Previous 1 2 Next

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
11	Salsa and Chips	Mexican	20	04/27/2020 12:00 AM	Moffet Field, Mountain View, CA 94035	Nuts, diary	None			OPEN

Showing 11 to 11 of 11 entries

Previous 1 2 Next

## 21. Search Engine Optimization

Search Engine Optimization is the process of improving internal and external aspects of a website or webpage to increase its visibility for search engines. It involves the website's HTML code and content to make it more search engine friendly and promoting the site to increase its relevance on the web. There are 10 techniques that can be used to achieve SEO. The following techniques have been employed in our application:

### 1. Title Tag

All pages have a title tag .

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title><%= msgs.title %></title>
```

### 2. H1 header Tag

Pages have a single H1 tag. This is slightly different from the title but have same meaning or context.

```
<h1>
  <%= msgs.shareYourFood %>
</h1>
```

### 3. Navigation:

Navigation is available in all pages. Header and side navigation are present in all dashboard pages, i.e. after the user logs in. Before logging in, header navigation is present in all pages.

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <!-- <div class="navbar-header">
      <a class="navbar-brand" href="/"></a>
    </div>-->
    <ul class="nav navbar-nav">
      <li><a class="navbar-brand" href="/"></a></li>
      <li><a href="/"><%= navLabels.home %></a></li>
      <li><a href="/problem"><%= navLabels.problem %></a></li>
      <li><a href="/donors"><%= navLabels.donate %></a></li>
      <li><a href="/shelters"><%= navLabels.shelters %></a></li>
      <li><a href="/volunteers"><%= navLabels.volunteer %></a></li>
    </ul>

    <ul class="nav navbar-nav navbar-right">
      <li><a href="/signin"><%= navLabels.signin %></a></li>
      <li><a href="/setting"><%= navLabels.settings %></a></li>
    </ul>
  </div>
</nav>
```

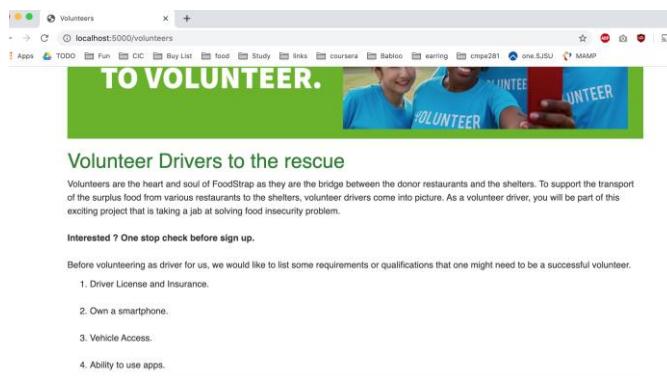
#### 4. Image Tag:

All images have a logical filename. The name denotes what the image is for. Alt attributes are also added to all img tags with keywords to provide more information.

```
<div class="container">
  <div class="item">
    
  </div>
```

#### 5. Page Content:

Short and crisp texts, no duplication of texts across pages. 300 to 700 words



#### 6. Description tag

Meta description tag is added to pages.

```
foodstrap > views > index.ejs > ...
1  <!DOCTYPE html> parvathysjsu [7 weeks ago] • initial files skeleton code
2  <html lang="en">
3
4  <head>
5    <title><%= msgs.title %></title>
6    <meta name="FoodStrap" content="Making it easy to donate food and bridging the gap between hunger and food waste.">
```

## 7. Rich snippets:

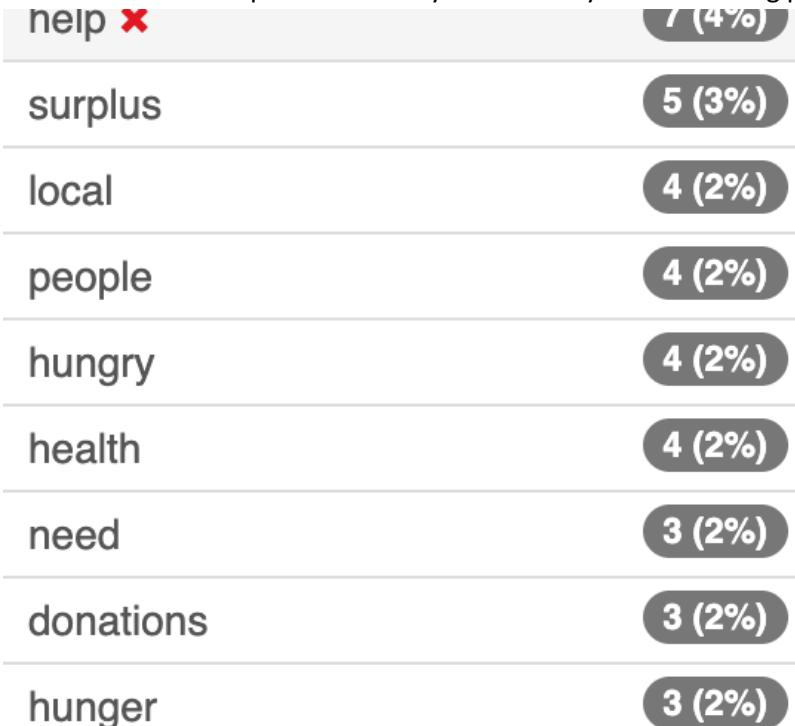
Using structured data format, we have enabled rich snippets in our application for our logo image. For a shelter or food donation, the address and phone number information is provided. Event location and time is present, and map shows the location.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Organization",
  "url": "http://ec2-3-101-63-213.us-west-1.compute.amazonaws.com:5000/",
  "logo": "http://ec2-3-101-63-213.us-west-1.compute.amazonaws.com:5000/img/logo.JPG"
}
```

## 8. Keyword Density:

Identified keywords for each page in the application and have used a variation of those keywords to avoid keyword stuffing.

Example 1: Below is the snapshot of the keyword density for the landing page.



Example 2: Snapshot of keyword density for the problem page



## 22. Profiling

### 01. Heap snapshot

Using Heap snapshot, memory distribution of JavaScript objects and related DOM nodes in the website was done. Summary view shows objects grouped by the constructor name. We can use it to check objects and track down DOM leaks. Containment view allows exploration of heap contents.

Constructor	Distance	Shallow Size	Retained Size
> Object	-	2 055 436	33 %
> Object	2	496 224	8 %
> Object	-	150 943	2 %
> Object	-	1 747 448	36 %
> Object	-	5 198 512	90 %
> Array	1	36	0 %
> Window	3	725 408	12 %
> (compiled code)	3	59 308	1 %
> system / Context	1	36	0 %
> Window / chrome-extension://kbfhbcaaplbcioakkpcsgfkobighlhen	2	116	0 %
> E	-	328 088	5 %
> Array	2	525 560	8 %
> (string)	2	525 560	8 %
> SVGSVGElement	4	208	0 %
> govt_JL	12	100	0 %
> Window	2	464	0 %
> (regepg)	2	6 524	0 %
> Module	9	388	0 %
> fe	14	112	0 %
> t <149	7	8 312	0 %
> Object /	1	20	0 %
> (concatenated string)	2	50 540	1 %
> SVGPPathElement	4	21 916	0 %
> Window / chrome-extension://gighmmplkbffepjocremgkkbigldom	1	36	0 %
> govt_Vk	3	632	0 %
> Window /	1	36	0 %
> govt_J	3	1 243	0 %
> govt_Ek	6	224	0 %
> govt_HM	19	124	0 %
> b	3	604	0 %
> Document	4	80	0 %
> e <103	8	3 256	0 %
> govt_Ak	3	84	0 %
> min_rnb <1	1	156	0 %

Object	Distance	Shallow Size	Retained Size
Object			

## FoodStrap Project Report

Memory tab selected.

**Profiles** section shows **HEAP SNAPSHOTS** and **Snapshot 1 Save**.

**Summary** and **Class filter** are applied.

**All objects** are listed in the main table.

The table has columns: Distance, Shallow Size, Retained Size.

**Constructor** section lists various object types and their counts:

Object Type	Count	Shallow Size	Retained Size
▶ (system)	×76121	2 055 436	33 %
▶ (closure)	×16206	496 224	8 %
▶ Object	×6647	150 948	2 %
▶ (array)	×25486	1 747 056	28 %
▶ Window / http://ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000	1	36	0 %
▶ (compiled code)	×13722	725 408	12 %
▶ system / Context	×1960	59 308	1 %
▶ Window / chrome-extension://kbfnbciaeplbcioakkpcpgfkobkgihlen	1	36	0 %
▶ E	2	116	0 %
▶ Array	×20495	328 088	5 %
▶ (string)	×21710	525 560	8 %
▶ SVGSVGElement	×8	208	0 %
▶ gvjs_8L	12	100	0 %
▶ Window	×17	464	0 %
▶ (regexp)	×233	6 524	0 %
▶ Module	×15	388	0 %
▶ fe	14	112	0 %
▶ t	×149	8 312	0 %
▶ Object /	1	20	0 %
▶ (concatenated string)	×2527	50 540	1 %
▶ SVGPathElement	×783	21 916	0 %
▶ Window / chrome-extension://gighmmpioblkfepjocnamgkkbiglidom	1	36	0 %
▶ gvjs_Vk	12	632	0 %
▶ Window /	1	36	0 %
▶ gvjs_J	×24	1 248	0 %
▶ gvjs_Ek	×4	224	0 %
▶ gvjs_HM	19	124	0 %
▶ b	×3	604	0 %
▶ Document	×4	80	0 %
▶ e	×103	3 256	0 %
▶ gvjs_4k	×3	84	0 %
▶ nvis_nB	×3	156	0 %

**Retainers** section lists objects:

Object	Distance	Shallow Size	Retained Size
Object			

Memory Tab - Object Containment

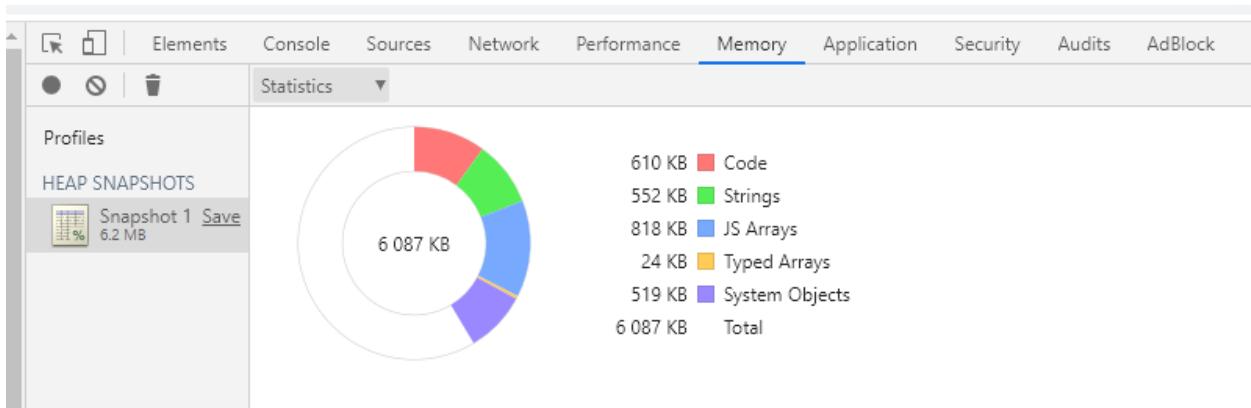
	Distance	Shallow Size	Retained Size
▼2 :: Window / ec2-3-101-76-67.us-west-2.amazonaws.com:5000 @4323 □	1	36 0 %	1 432 584 23 % ▲
►properties :: (object properties[]) @188585	2	32 796 1 %	135 260 2 %
►native_context :: system / NativeContext @4321	2	1 060 0 %	133 676 2 %
►\$ :: k() @46989 □	2	32 0 %	40 944 1 %
►jQuery :: k() @46989 □	2	32 0 %	40 944 1 %
►gvjs_Ol :: Object @154939 □	2	28 0 %	13 700 0 %
►gvjs_Baa :: /(?!rgb)?((\([1-9]\d{0,2}\),\s?(\([1-9]\d{0,2}\),\s?(\([1-9]\d{0,2}\),\s?(\([1-9]\d{0,2}\))\)\$)/	2	28 0 %	12 844 0 %
►gvjs_Wba :: gvjs_5j @185159 □	2	76 0 %	10 188 0 %
►gvjs_RM :: Object @155445 □	2	20 0 %	10 108 0 %
►gvjs_5c :: Object @167185 □	2	12 0 %	8 312 0 %
►gvjs_Ud :: Object @167257 □	2	248 0 %	7 384 0 %
►gvjs_Oj :: gvjs_0J() @153817 □	2	32 0 %	6 932 0 %
►gvjs_Q :: gvjs_Q() @95241 □	2	32 0 %	6 832 0 %
►gvjs_Jba :: Object @184433 □	2	400 0 %	6 304 0 %
►<symbol> :: HTMLDocument @15221 □	2	28 0 %	5 788 0 %
►gvjs_Dba :: Object @188273 □	2	348 0 %	5 588 0 %
►google :: Object @77597 □	2	28 0 %	5 392 0 %
►gvjs_nm :: gvjs_nm() @95269 □	2	32 0 %	5 324 0 %
►gvjs_OF :: Object @152753 □	2	112 0 %	4 952 0 %
►gvjs_jd :: gvjs_jd() @138225 □	2	32 0 %	4 876 0 %
►gvjs_O :: Object @95309 □	2	16 0 %	4 656 0 %
►gvjs_jA :: gvjs_jA() @149559 □	2	32 0 %	4 648 0 %
►gvjs_Uca :: gvjs_E @184063 □	2	16 0 %	4 332 0 %
►gvjs_UB :: gvjs_UB() @150359 □	2	32 0 %	4 304 0 %
►gvjs_Oz :: gvjs_Oz() @149311 □	2	32 0 %	4 192 0 %
►gvjs_lz :: gvjs_lz() @149437 □	2	32 0 %	4 164 0 %
►gvjs_6C :: gvjs_6C() @151247 □	2	32 0 %	3 824 0 %
►gvjs_3l :: Object @155347 □	2	92 0 %	3 812 0 %
►gvjs_Lca :: Object @184935 □	2	108 0 %	3 712 0 %
►gvjs_fd :: Object @167199 □	2	104 0 %	3 664 0 %
►gvjs_gd :: Object @167199 □	2	104 0 %	3 664 0 %
►gvjs_Eda :: Object @149727 □	2	172 0 %	3 324 0 %
►gvjs_1m :: Object @186229 □	2	200 0 %	3 312 0 % ▾

	Distance	Shallow Size	Retained Size
Object			
►extension in system / NativeContext @4321	2	1 060 0 %	133 676 2 % ▲
►native_context in Window / ec2-3-101-76-67.us-west-2.amazonaws.com:5000 @4323 □	1	36 0 %	1 432 584 23 %
►context in () @188081 □	2	28 0 %	164 0 %
►context in () @187251 □	2	28 0 %	164 0 %
►context in () @186765 □	2	28 0 %	164 0 %
►context in () @167161 □	2	32 0 %	100 0 %
►context in () @167133 □	2	32 0 %	140 0 %
►context in () @167101 □	2	32 0 %	100 0 % ▾

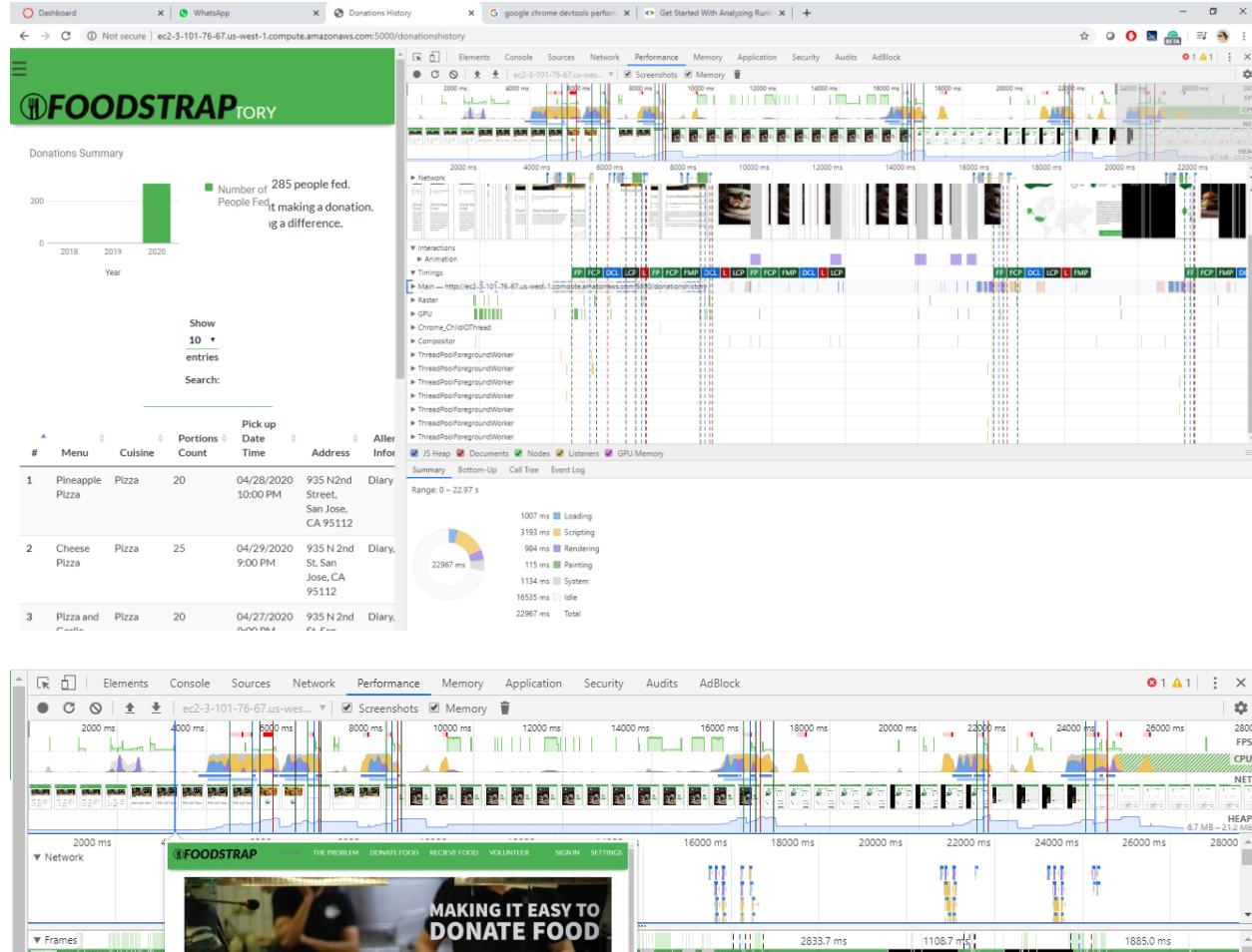
Console Tab - What's New

Statistics tab shows how much memory each type of object that is present takes up overall.

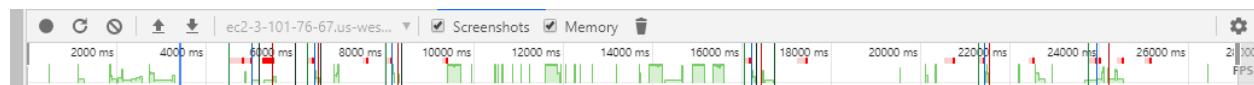


## 02. Performance Profiling

Using the Performance panel in Devtools in Chrome, the website was profiled, to find a performance bottleneck. The runtime performance was recorded for the restaurant manager flow. It includes the landing page, problems page, login of restaurant manager, manager dashboard, donate and history pages.

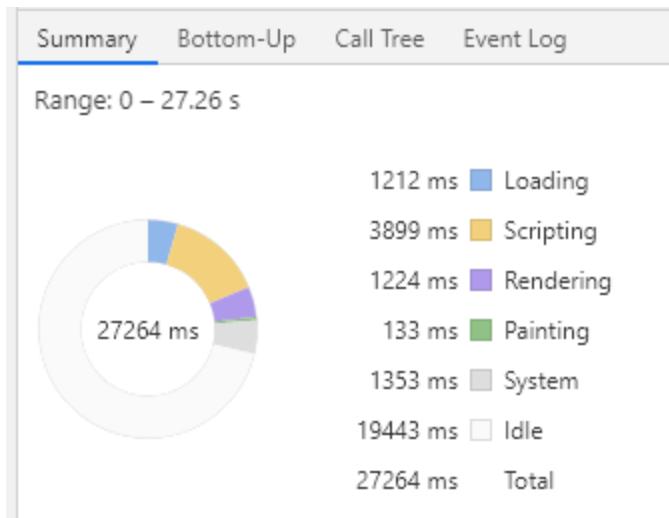


The main metric for measuring the performance of any animation is frames per second (FPS). As shown in the **FPS** chart in the image, occurrence of red bars is very rare meaning the framerate didn't drop very low. The green bar shows that FPS is high.



As seen in the CPU chart, utilization was low and CPU was not maxed out. The same CPU profile is shown in the summary tab in a graph.





NET and HEAP tabs results also shown in image.

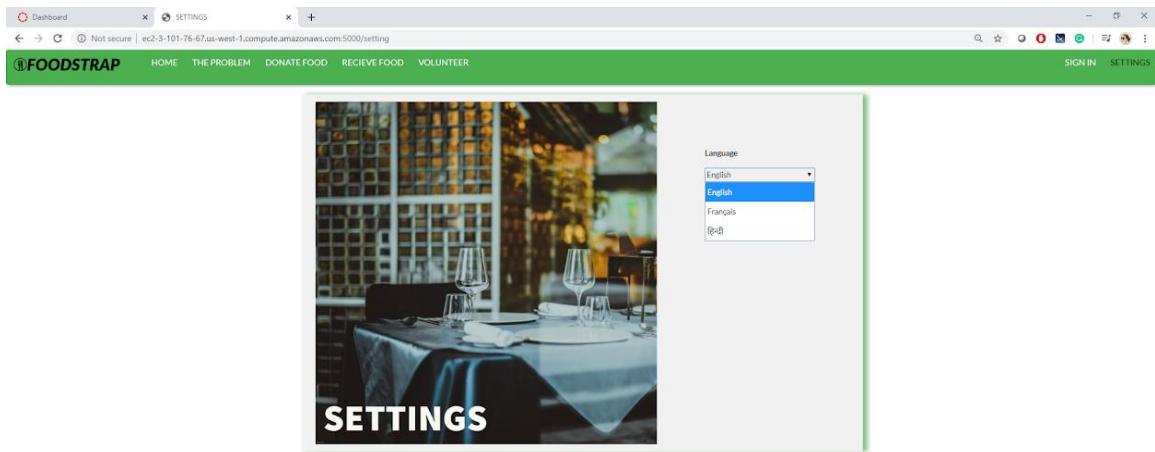


## 23. Localization

Website localization is the process of adapting an existing website to local language and culture in the target market. For this application, we have done localization for Hindi and French. In the Settings Page, the user can set the desired language. Once the language is set, the website is shown in that language.

Screenshots of the website in Hindi and French are added in the following sections:

- Hindi: Screenshots section > Localization > Hindi
- French: Screenshots section > Localization > French





**SIGN IN**

Help eliminate food waste  
and feed the hungry.

Your username \_\_\_\_\_  
Your password \_\_\_\_\_  
User Type \_\_\_\_\_  
  
SIGN IN  
  
New User? SIGN UP

**TEXT AND IMAGE IN ENGLISH**



**साइन इन**

भोजन की बर्बादी को खत्म करने में मदद और  
भूखे को खाना खिलाएं।

आपका उपयोगकर्ता नाम \_\_\_\_\_  
आपका पासवर्ड \_\_\_\_\_  
उपयोगकर्ता का प्रकार \_\_\_\_\_  
  
साइन इन करें  
  
नया उपयोगकर्ता? साइन अप करें

**TEXT AND IMAGE IN HINDI**

Localization is implemented using a web resource json file and all the text and image names for all languages are in that file. According to the language set, the text and images are fetched from the json file and sent as parameters to the ejs file and displayed in UI.

Localization json file: FoodStrap\foodstrap\public\settings\localization.json

```

foodstrap > public > settings > localization.json > page > index > en
1  {
2    "applicationName": "foodstrap",
3    "page": {
4      "index": {
5        "en": [
6          {
7            "title": "FoodStrap",
8            "quote": "If you can't feed a hundred people, then feed just one!", | parvathysjsu [5 weeks ago] • Localization for index page
9            "quoteAuthor": "Mother Teresa",
10           "descWhatAreWe": "What are we?",
11           "descWhatWeAnswer": "We provide an alternative to discarding surplus wholesome food by linking food service donors with surplus food to local hunger relief organizations.",
12           "descWhatWeDo": "What do we do to help the community?",
13           "descWhatWeDoAnswer": "We help people who are having a financial crisis. They may have lost their job, or had a medical emergency, or their car may have broken down, or they may be unable to afford food for their family.", | parvathysjsu [5 weeks ago] • Localization for index page
14           "descFromWhereFood": "From where do we get our food?",
15           "descFromWhereFoodAnswer": "They have few sources from where we get the food. It can be restaurants who have surplus food, caterers who have leftover food, or individuals who have excess food.", | parvathysjsu [5 weeks ago] • Localization for index page
16           "shareYourFood": "Share Your Food",
17           "shareYourFoodAnswer": "Many families in our area need a little help putting food on the table. They shouldn't have to choose between paying rent or buying food.", | parvathysjsu [5 weeks ago] • Localization for index page
18           "donateFood": "Donate Surplus Food",
19           "donateFoodAnswer": "Instead of throwing away surplus food that is safe to eat, donate to local hunger relief organizations. This delivers a triple benefit: it helps those in need, it reduces food waste, and it supports local businesses.", | parvathysjsu [5 weeks ago] • Localization for index page
20           "receiveFood": "Receive Food",
21           "receiveFoodAnswer": "The quality, wholesome prepared food donated from restaurants can serve those who experience hunger. If you have to choose between paying rent or buying food, choose food.", | parvathysjsu [5 weeks ago] • Localization for index page
22           "caption_map": "Portions Donated per Country"
23         },
24         "fr": [
25           {
26             "title": "FoodStrap",
27             "quote": "Si vous ne pouvez pas nourrir une centaine de personnes, n'en nourrissez qu'une!", | parvathysjsu [5 weeks ago] • Localization for index page
28             "quoteAuthor": "Mère Teresa",
29             "descWhatAreWe": "Que sommes-nous?", | parvathysjsu [5 weeks ago] • Localization for index page
30             "descWhatWeAnswer": "Nous offrons une alternative à l'élimination des surplus d'aliments sains en reliant les donneurs de services alimentaires avec les organisations de relais de nourriture.", | parvathysjsu [5 weeks ago] • Localization for index page
31             "descWhatWeDo": "Que faisons-nous pour aider la communauté?", | parvathysjsu [5 weeks ago] • Localization for index page
32             "descWhatWeDoAnswer": "Nous aidons les personnes qui traversent une crise financière. Ils ont peut-être perdu leur emploi, eu une urgence médicale ou leur maison a été évacuée.", | parvathysjsu [5 weeks ago] • Localization for index page
33             "descFromWhereFood": "D'où obtenons-nous notre nourriture?", | parvathysjsu [5 weeks ago] • Localization for index page
34             "descFromWhereFoodAnswer": "Ils ont peu de sources d'où nous obtenons la nourriture. Il peut s'agir de restaurants qui ont un surplus de nourriture, de traiteurs ou de fournisseurs.", | parvathysjsu [5 weeks ago] • Localization for index page
35             "shareYourFood": "Partagez votre nourriture", | parvathysjsu [5 weeks ago] • Localization for index page
36             "shareYourFoodAnswer": "De nombreuses familles de notre région ont besoin d'un peu d'aide pour mettre de la nourriture sur la table. Ils ne devraient pas avoir à faire de choix entre la nourriture et autre chose.", | parvathysjsu [5 weeks ago] • Localization for index page
37             "donateFood": "Donnez un surplus de nourriture", | parvathysjsu [5 weeks ago] • Localization for index page
38             "donateFoodAnswer": "Au lieu de jeter les surplus de nourriture qui sont sûrs à manger, faites un don aux organisations locales de lutte contre la faim. Cela aide à réduire le gaspillage et à soutenir les communautés.", | parvathysjsu [5 weeks ago] • Localization for index page
39             "receiveFood": "Recevoir de la nourriture", | parvathysjsu [5 weeks ago] • Localization for index page
40             "receiveFoodAnswer": "La nourriture saine et de qualité préparée donnée par les restaurants peut servir ceux qui ont faim. Si vous devez choisir entre payer une facture ou acheter de la nourriture, choisissez la nourriture.", | parvathysjsu [5 weeks ago] • Localization for index page
41             "caption_map": ""
42           },
43           "hi": [
44             {
45               "title": "फूडस्ट्रप",
46               "quote": "आप सी लोगों को नहीं हिला सकते हैं, तो सिर्फ एक को हिलाएं", | parvathysjsu [5 weeks ago] • Localization for index page
47               "quoteAuthor": "मदर टेरेसा",
48               "descWhatAreWe": "हम क्या हैं?", | parvathysjsu [5 weeks ago] • Localization for index page
49               "descWhatWeAnswer": "हम खाव सेवा गताओं को अधिशेष भोजन के साथ स्थानीय भूख राहत एजेंसियों से जोड़कर अधिशेष प्रौद्योगिक भोजन छोड़ने का विकल्प प्रदान करते हैं। हम एक कुशल संचार और समर्पित सेवा गताओं की मदद के लिए क्या करते हैं?", | parvathysjsu [5 weeks ago] • Localization for index page
50               "descWhatWeDo": "हम समदाय की मदद के लिए क्या करते हैं?", | parvathysjsu [5 weeks ago] • Localization for index page
51             }
52           ]
53         }
54       }
55     }
56   }
57 }
```

## 01. Text Localization

The process for displaying a text in a particular language:

- Find the language set from the constants variable.
- For each text, there is a unique key that is used to identify the text.
- For each page, there is a key that is used to identify the page.
- Get all the messages for a particular page and particular language.
- Pass this to the ejs file.
- In the ejs file, from this variable that has all the messages, get a particular message using its key and display in UI.

### Example

## Username in Sign In page

- JSON file: FoodStrap\foodstrap\public\settings\localization.json
- Page key: signin
- Text key: username

1. Translations for this in various languages are present in the json file.
2. In route.js, first get the language set from the constant variable.

```
var lang = constants.properties.lang;
```

3. Then get the messages for sign page and language set

```
var msgsVar = messages.page.signin[lang];
```

4. Pass it to ejs file

```
res.render('signin', {
  msgs: msgsVar,
  --
})
```

5. In ejs file, get the text from msgs variable and the key for username text.

```
<label for="username" class="form-label"><%= msgs.username %></label>
```

```

foodstrap > public > settings > {} localization.json > {} page > {} signin > {} hi
499     }
500   },
501   "signin": {
502     "en": {
503       "username": "Your username",
504       "pwd": "Your password",
505       "usertype": "User Type",
506       "rest": "Restaurant",
507       "shelter": "Shelter",
508       "vol": "Volunteer",
509       "signin": "SIGN IN",
510       "signup": "New User? SIGN UP"
511     },
512     "fr": {
513       "username": "Your username",
514       "pwd": "Your password",
515       "usertype": "User Type",
516       "rest": "Restaurant",
517       "shelter": "Shelter",
518       "vol": "Volunteer",
519       "signin": "SIGN IN",
520       "signup": "New User? SIGN UP"
521     },
522     "hi": [
523       "username": "आपका उपयोगकर्ता नाम",
524       "pwd": "आपका पासवर्ड",
525       "usertype": "उपयोगकर्ता का प्रकार",
526       "rest": "भोजनालय",
527       "shelter": "आश्रय",
528       "vol": "वालंटियर",
529       "signin": "साइन इन करें",
530       "signup": "नया उपयोगकर्ता? साइन अप करें"
531     ]
532   },
533 }

```

```

foodstrap > routes > js route.js > ...
142   });
143   /* show signin page */
144   router.get('/signin', function (req, res, next) {
145     var lang = constants.properties.lang;
146     console.log(lang);
147     var msgsVar = messages.page.signin[lang];
148     res.render('signin', {
149       msgs: msgsVar,
150       navLabels: messages.page.nav[lang],
151       imgNames: messages.page.images[lang]
152     });
153   });

```

The screenshot shows a code editor with the following file structure:

```

foodstrap > views > signin.ejs > html > body > div.container.greyShadow > div > div.sectionDivider35 > form#signup-form
47   </div>
48   <div class="sectionDivider35">
49   <form id="signup-form" action="/signin" method="post">
50     <div class="form-group form-input">
51       <input type="text" name="username" id="username" value="" required />
52       <label for="username" class="form-label"><%= msgs.username %></label>
53     </div>

```

The code is part of a sign-in form. It includes a form with an ID of "signup-form" and an action of "/signin". Inside the form, there is a text input field with the name "username" and an associated label. The label contains a placeholder from a localization JSON file.

## 02. Image Localization

For each language, along with text, images also should be different to be sensitive to that locale. Using a web resource file to store image names for different languages, localization for images is done.

The process for displaying an image in a particular language:

- Find the language set from the constants variable.
- For each image, there is a unique key that is used to identify that image.
- Get all the image names for a particular language.
- Pass this to the ejs file.
- In the ejs file, from this variable that has all the image names, get a particular image name for a language using its key and display in UI.

### Example

Sign In image in Sign In page

- JSON file: FoodStrap\foodstrap\public\settings\localization.json
- Images key: images
- Sign in image key: img\_signin

1. Translations for this in various languages are present in the json file.
2. In route.js, first get the language set from the constant variable.

```
var lang = constants.properties.lang;
```

3. Then get the image names for language set

```
messages.page.images[lang]
```

4. Pass it to ejs file

```
res.render('signin', {
  -
  -
  imgNames: messages.page.images[lang]
})
```

5. In ejs file, get the image name from imgNames variable and the key for image name.

```

```

```

foodstrap > public > settings > {} localization.json > {} page > {} images
645     },
646     "images":[]  parvathysjsu [4 weeks ago] • Localization for images
647     "en": {
648         "img_foodstrap":"img/foodstrap_banner.jpg",
649         "img_problem":"img/problem.jpg",
650         "img_donate":"img/donatefood.jpg",
651         "img_rcv":"img/recievefood.jpg",
652         "img_vol":"img/volunteer.jpg",
653         "img_signin":"img/signin.jpg",
654         "img_settings":"img/settings.jpg",
655         "img_signup":"img/signup.jpg",
656         "img_problemhead":"img/problem_header.jpg",
657         "img_volhead":"img/vol2.jpg",
658         "img_VOLDASH":"img/VOL_DASH_EN.jpg",
659         "img_RESTDASH":"img/restdashbanner_en.jpg",
660         "img_shDASH":"img/SH_DASH_EN.jpg"
661     },
662     "fr": {
663         "img_foodstrap":"img/foodstrap_banner.jpg",
664         "img_problem":"img/problem.jpg",
665         "img_donate":"img/donatefood.jpg",
666         "img_rcv":"img/recievefood.jpg",
667         "img_vol":"img/volunteer.jpg",
668         "img_signin":"img/signin.jpg",
669         "img_settings":"img/settings.jpg",
670         "img_signup":"img/signup.jpg",
671         "img_problemhead":"img/probf.png",
672         "img_volhead":"img/volfrance.png"
673     },
674     "hi": {
675         "img_foodstrap":"img/foodstrap_banner_hi.jpg",
676         "img_problem":"img/problem.jpg",
677         "img_donate":"img/donatefood_hi.jpg",
678         "img_rcv":"img/recievefood_hi.jpg",
679         "img_vol":"img/volunteer.jpg",
680         "img_signin":"img/signin_hi.jpg",
681         "img_settings":"img/settings_hi.jpg",
682         "img_signup":"img/signup_hi.jpg",
683         "img_problemhead":"img/probh.png",
684         "img_volhead":"img/volhindi.png",
685         "img_VOLDASH":"img/VOL_DASH_hi.jpg",
686         "img_RESTDASH":"img/restdashbanner_HI.jpg",
687         "img_shDASH":"img/SH_DASH_hi.jpg"
688     }
689 }

```

```

foodstrap > routes > js route.js > Y router.get('/signin') callback > Y imgNames
142   });
143   /* show signin page */
144   <router.get('/signin', function (req, res, next) {
145     var lang = constants.properties.lang;
146     console.log(lang);
147     var msgsVar = messages.page.signin[lang];
148     res.render('signin', {
149       msgs: msgsVar,
150       navLabels: messages.page.nav[lang],
151       imgNames: messages.page.images[lang]      parvathysjsu
152     });
153   });

```

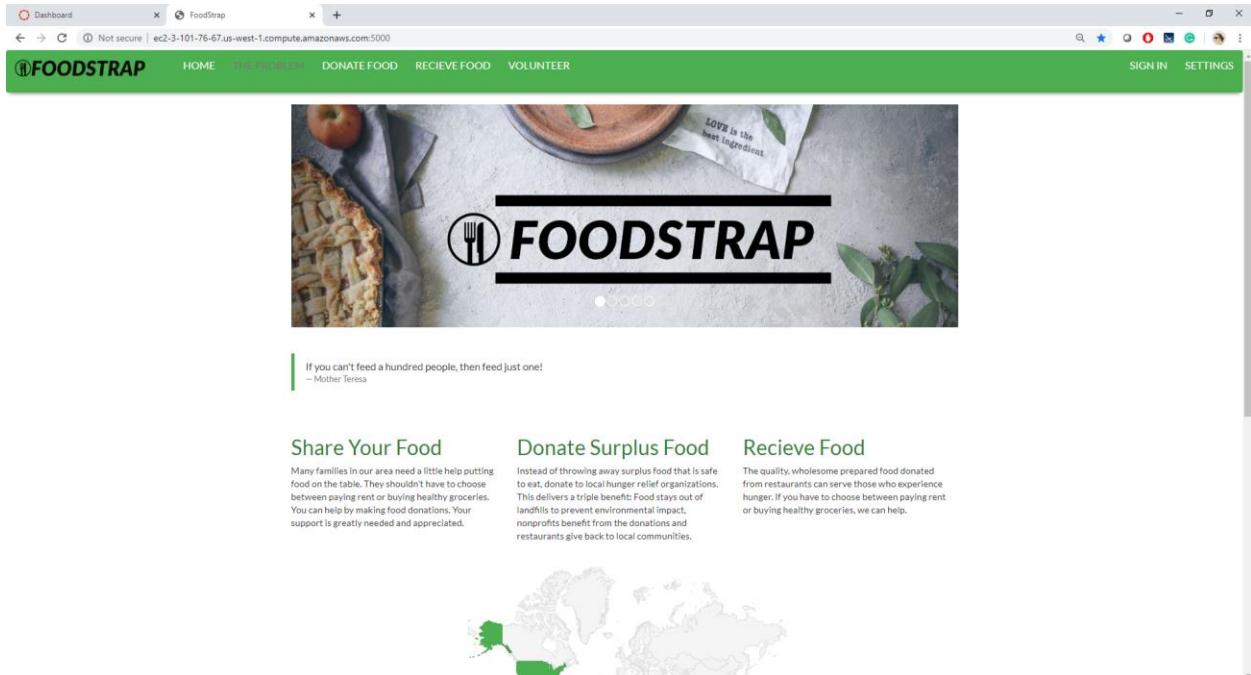
```

foodstrap > views > < signin.ejs > Y html > Y body > Y div.container.greyShadow > Y div > Y div > Y div
44   <div>
45     <div class="sectionDivider65" style="background-color: #f0f0f0; height: 1px; margin-bottom: 10px; position: relative; z-index: 1; border: none; width: 100%;"> parvathysjsu [6 weeks ago]
46       
47     </div>

```

## 24. Screenshots

### 01. Application



You can help by making food donations. Your support is greatly needed and appreciated.

landfills to prevent environmental impact, nonprofits benefit from the donations and restaurants give back to local communities.

Portions Donated per Country

**What are we?**  
We provide an alternative to discarding surplus wholesome food by linking food service donors with surplus food to local hunger relief agencies. We do this by creating and maintaining an efficient communication and reporting network that links available sources of food to those in need through these existing charitable organizations.

**What do we do to help the community?**  
We help people who are having a financial crisis. They may have lost their job, or had a medical emergency, or their car may have broken down. We help the people who are needy or who are poor and need some help getting food. People that go hungry for a long period of time will change as a person completely. More than seventy percent of people go hungry in the world. Going hungry for a certain period of time damages your mental health and physical health. When it damages your mental health it causes you to have negative thoughts about things. Going hungry also changes your physical health, it causes you to lose weight and it will mess up your respiratory system.

**From where do we get our food?**  
They have few sources from where we get the food. It can be restaurants who have surplus food, caterers who have leftover food, individual donations, local businesses around the area.

© 2020 Created by Parvathy & Mayura for CMPE280 Web UI Design

**FOODSTRAP**

HOME THE PROBLEM DONATE FOOD RECEIVE FOOD VOLUNTEER SIGN IN SETTINGS

THE PROBLEM

Food Insecurity

people us affordable  
financial  
**insecurity**  
poverty resources  
wages **Food** problem  
lack  
needs

The world currently produces more than enough food to feed everyone, yet 815 million people (roughly 11% of the global population) went hungry In 2016, according to the U.N. The U.S. Department of Agriculture (USDA) defines food insecurity as a lack of consistent access to enough food for an active, healthy life.

Food Waste.

# FoodStrap Project Report

The screenshot displays two views of a web application. The top view shows the homepage with a banner about food donation, a quote from Mahatma Gandhi, and a map of the San Francisco Bay Area. The bottom view shows a detailed map of the same region with specific locations highlighted.

**MAKING IT EASY TO DONATE FOOD**

The simplest acts of kindness are by far more powerful than a thousand heads bowing in prayer.  
— Mahatma Gandhi

Many families in our area need a little help putting food on the table. They shouldn't have to choose between paying rent or buying healthy groceries. You can help by making food donations. Your support is greatly needed and appreciated. In the US, food waste is estimated at between 30-40% of our food supply, estimated at around 72 billion-133 billion pounds of wasted food each year, according to the USDA. Rather than toss unsold breads, pastries, and produce to a dumpster, box them up for local nonprofits and those in need. From the Michelin-starred, national restaurant chains to world-wide mega coffee shops that proliferate every street corner in America, check out these establishments that are committed to cut down on food waste by donating at the end of the day.

**SIGN UP NOW!**



## Receive Food Assistance

The simplest acts of kindness are by far more powerful than a thousand heads bowing in prayer.  
— Mahatma Gandhi

Last year, almost 41 million tons of food were generated in the United States. While Americans dispose of millions of tons of food, the U.S. Department of Agriculture estimates that 11.8 percent of American households - about 15 million households - had difficulty providing enough food for all their members due to a lack of resources. In many cases, the food tossed into our nation's landfills is wholesome, edible food.

If you need food assistance for your shelter, sign up now!

[SIGN UP NOW!](#)



## Volunteer Drivers to the rescue

Volunteers are the heart and soul of FoodStrap as they are the bridge between the donor restaurants and the shelters. To support the transport of the surplus food from various restaurants to the shelters, volunteer drivers come into picture. As a volunteer driver, you will be part of this exciting project that is taking a jab at solving food insecurity problem.

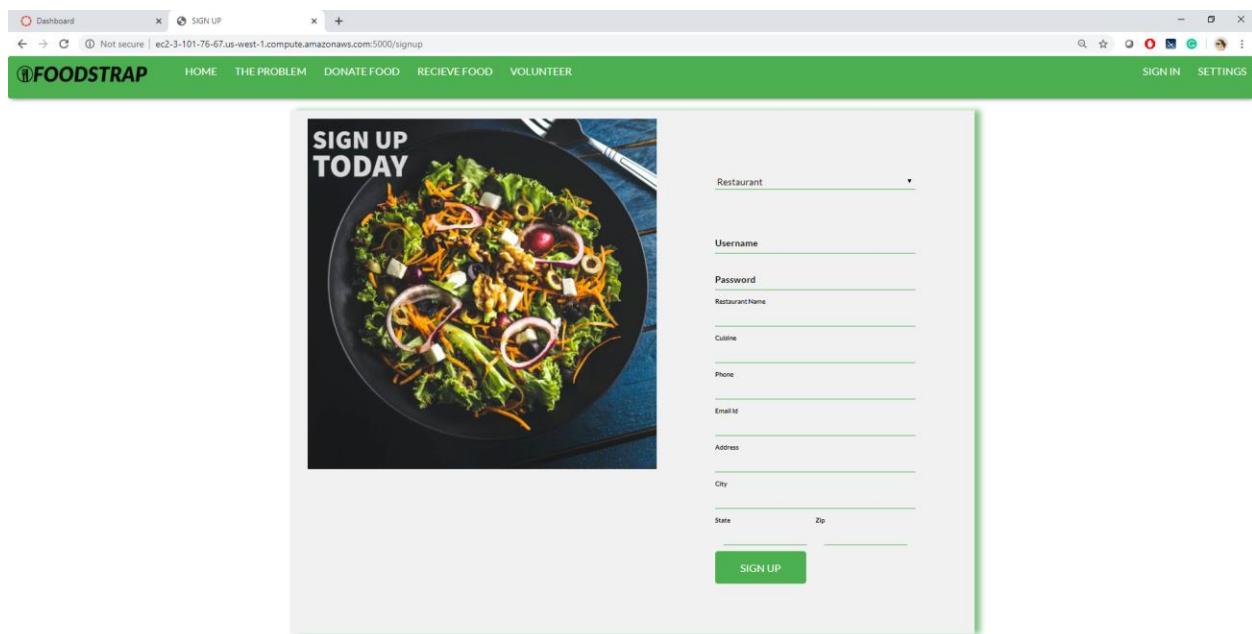
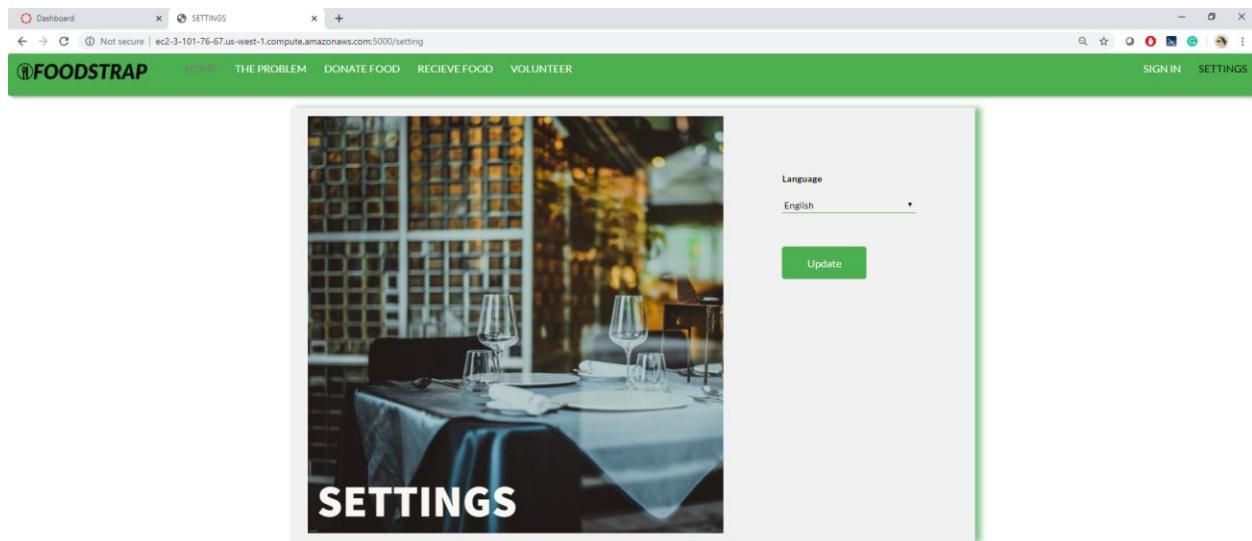
Interested? One stop check before sign up.

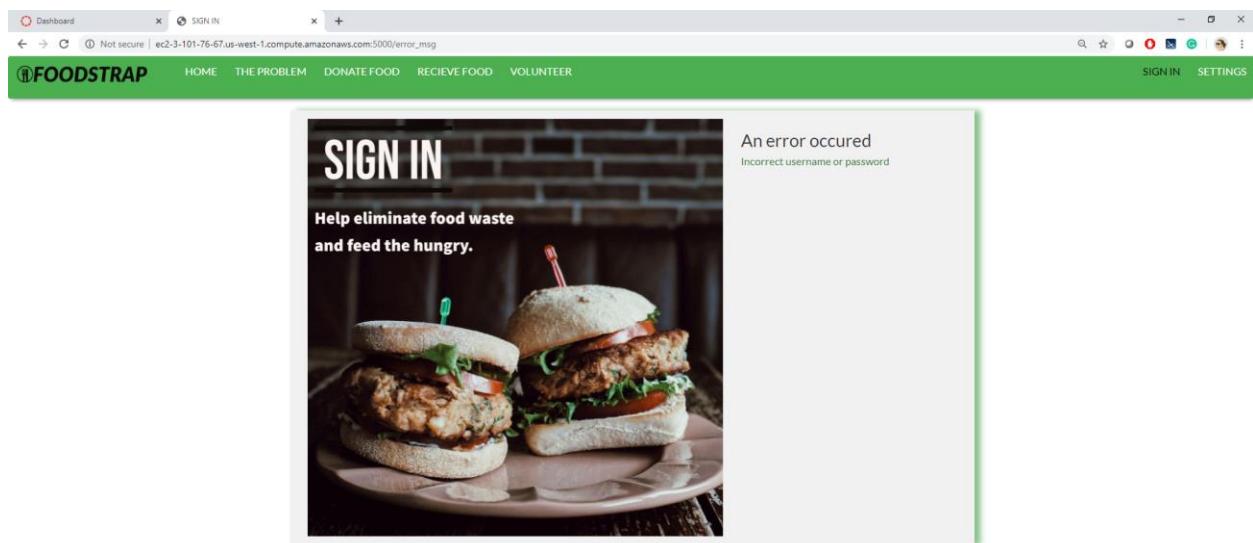
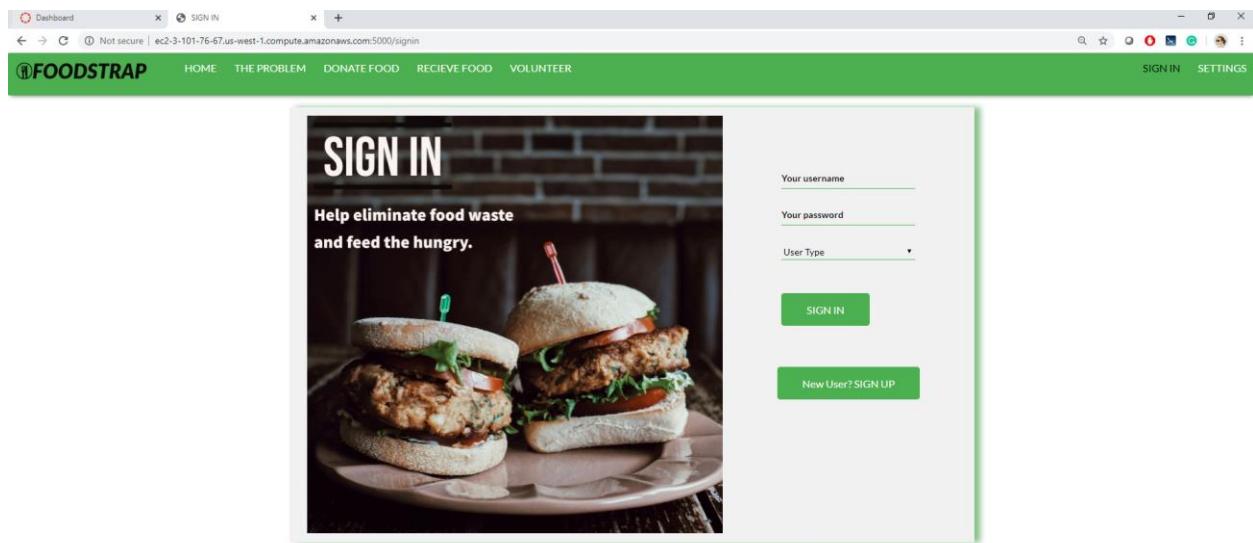
Before volunteering as driver for us, we would like to list some requirements or qualifications that one might need to be a successful volunteer.

1. Driver License and Insurance.
2. Own a smartphone.
3. Vehicle Access.
4. Ability to use apps.

Are you Interested?  
[Sign Up Now!](#)

## FoodStrap Project Report





The screenshot shows a web browser window titled "Dashboard" with the URL "Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/restaurant\_dashboard". The page has a green header bar with the text "RESTAURANT MANAGER DASHBOARD" and the "FOODSTRAP" logo. Below the header is a large green icon of a hand holding a tray with food items. To the right of the icon, the text "BE A PART OF THE SOLUTION..." is displayed above the heading "DONATE SURPLUS FOOD". Below this heading are six input fields with icons: DATE (calendar), LOCATION (location pin), NOTES (list), TIME (clock), PORTION COUNT (plate), and MENU (trash can). To the right of the map, there is a text block about food waste in the U.S. and a green button labeled "Donate Today and Save Lives!".

This screenshot shows the same dashboard interface as the first one, but with a vertical black sidebar on the left containing navigation links: "DASHBOARD", "DONATE", "HISTORY", "PROCESS", "PROFILE", and "LOG OUT". The rest of the page content is identical to the first screenshot, including the "DONATE SURPLUS FOOD" section and the world map with a callout for the United States.

**Dashboard**

Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/donate

**DONATE**

**FOODSTRAP**

**Menu**

Cuisine:  Portion Count:

Allergy Information:

Special Instructions:

Pick Up Time:

Pick Up Address:

City:  State:  Zip:

**Donate**

**Donations History**

Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/donationshistory

**DONATIONS HISTORY**

**FOODSTRAP**

Donations Summary

■ Number of People Fed

Year	Number of People Fed
2018	~10
2019	~100
2020	~200

Congratulations! 285 people fed.  
Giving is not just about making a donation. It's about making a difference.

Show 10 entries

Search:

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
1	Pineapple Pizza	Pizza	20	04/28/2020 10:00 PM	935 N 2nd Street, San Jose, CA 95112	Diary	Curb side pick up	Home For Good	Patrick Jane	DONE
2	Cheese Pizza	Pizza	25	04/29/2020 9:00 PM	935 N 2nd St, San Jose, CA 95112	Diary, Gluten	Curb side pick up	Home For Good	Patrick Jane	DONE
3	Pizza and Garlic Bread	Pizza	20	04/27/2020 9:00 PM	935 N 2nd St, San Jose, CA 95112	Diary, Gluten	None	Home For Good	Patrick Jane	DONE
4	Veg Pasta	Italian	20	04/26/2020 10:00 PM	935 N 2nd St, San Jose, CA 95112	Diary, nuts and gluten	None	Home For Good	Patrick Jane	DONE
5	Pizza and Garlic Bread	Pizza	35	04/29/2020 10:00 PM	935 N 2nd Street, San Jose, CA 95112	Diary, nuts and gluten	Curb side pick up	Home For Good	Patrick Jane	DONE
6	Veg Pasta	Italian	25	04/29/2020 9:00 PM	935 N 2nd Street, San Jose, CA 95112	Nuts, diary	None	Home For Good	Patrick Jane	DONE
7	Fried Pasta	Italian	30	04/30/2020 10:00 PM	935 N 2nd Street, San Jose, CA 95112	Nuts, diary	None	OPEN		

# FoodStrap Project Report

**DONATIONS HISTORY**

Donations Summary

Congratulations! 285 people fed.  
Giving is not just about making a donation. It's about making a difference.

Show	10 entries	Search:								
#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
7	Fried Pasta	Italian	30	04/30/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Nuts, diary	None			OPEN
12	Burritos and enchiladas	Mexican	30	05/07/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Nuts	Curb side pickup			OPEN

Showing 1 to 2 of 2 entries (filtered from 13 total entries)

Previous 1 Next

**DONATIONS PROCESS**

identify the food to be donated.  
↓  
Safely package and label food.  
Also note allergy information and special instructions if any.  
↓  
Properly chill, then freeze packaged food.  
↓  
Document the donation on Foodstrap providing relevant information.  
↓  
Hold for pick up.  
↓  
A volunteer or shelter claims the donation.  
↓  
The volunteer picks up the food and people are fed.

**PROFILE**

SAM PIZZERIA

Cuisine	Italian		
Phone	8885554445		
Email Id	sampspizza@gmail.com		
Address	935		
City	San		
State	CA	Zip	95112

**UPDATE**

**SHELTER DASHBOARD**

**CLAIM DONATIONS FROM RESTAURANTS**

NEED FOOD ASSISTANCE?

WHEN DO YOU NEED

HOW MANY YOU NEED

**Map** **Satellite**

Feed the Need  
Food is national security. Food is economy. It is employment, energy, history. Food is everything. – José Andrés

Food Insecurity has been on the rise and hunger is closely related to it. To help address this growing problem, organizations such as homeless shelters, food banks and soup kitchens are doing their part by feeding the hungry. With the aid of volunteers.

Our Location  
900 N1st Street, San Jose, CA - 95112

Our Volunteers

1	Patrick Jane
2	Teresa Lisbon

**SHELTER DASHBOARD**

**CLAIM DONATIONS FROM RESTAURANTS**

NEED FOOD ASSISTANCE?

WHEN DO YOU NEED

HOW MANY YOU NEED

**Map** **Satellite**

Feed the Need  
Food is national security. Food is economy. It is employment, energy, history. Food is everything. – José Andrés

Food Insecurity has been on the rise and hunger is closely related to it. To help address this growing problem, organizations such as homeless shelters, food banks and soup kitchens are doing their part by feeding the hungry. With the aid of volunteers.

Our Location  
900 N1st Street, San Jose, CA - 95112

Our Volunteers

1	Patrick Jane
2	Teresa Lisbon

# FoodStrap Project Report

Dashboard x FoodStrap x + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/shelter\_claim

CLAIM DONATIONS

FOODSTRAP

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Status	
1	Fried Pasta	Italian	30	04/30/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Nuts, diary	None	OPEN	<button>CLAIM</button>
2	Salsa and Chips	Mexican	40	04/30/2020 11:00 PM	Moffet Field, Mountain View, CA 94035	nuts	None	OPEN	<button>CLAIM</button>
3	Burritos and enchiladas	Mexican	30	05/07/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Nuts	Curb side pickup	OPEN	<button>CLAIM</button>

Show 10 entries

Search:

Showing 1 to 3 of 3 entries

Previous 1 Next

Dashboard x hello x + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/claim/Seab6b35f7fb7b4021e75f

CLAIM DONATION

Map Satellite

Menu: Fried Pasta  
Cuisine: Italian  
Portion Count: 30  
Allergy Information: Nuts, diary  
Special Instructions: None  
Pick Up Time: 04/30/2020 10:00 PM  
Pick Up Address: 935 N2nd Street, San Jose, CA 95112  
Status: OPEN

CANCEL CONFIRM CLAIM

# FoodStrap Project Report

Dashboard x FoodStrap x + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/past\_claim

## CLAIMS HISTORY

**Success Summary**

Number of People Fed

Congratulations! 310 people fed.  
Service to others is the rent you pay for your room here on earth.-- Mohammad Ali

Show 10 entries

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
1	Pineapple Pizza	Pizza	20	04/28/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Diary	Curb side pick up	Home For Good	Patrick Jane	DONE
2	Cheese Pizza	Pizza	25	04/29/2020 9:00 PM	935 N 2nd St. San Jose, CA 95112	Diary, Gluten	Curb side pick up	Home For Good	Patrick Jane	DONE
3	Pizza and Garlic Bread	Pizza	20	04/27/2020 9:00 PM	935 N 2nd St. San Jose, CA 95112	Diary, Gluten	None	Home For Good	Patrick Jane	DONE
4	Veg Pasta	Italian	20	04/26/2020 10:00 PM	935 N 2nd St. San Jose, CA 95112	Diary, nuts and gluten	None	Home For Good	Patrick Jane	DONE
5	Tacos	Mexican	30	04/29/2020 9:00 PM	108 Castro St., Mountain View, CA 94035	Nuts	None	Home For Good	Patrick Jane	DONE
6	Burrito	Mexican	20	04/29/2020 10:00 PM	Moffet Field, Mountain View, CA 94035	Nuts, diary	None	Home For Good	Patrick Jane	DONE
7	Tacos and Chips	Mexican	45	04/28/2020 9:00 PM	Moffet Field, Mountain View, CA 94035	nuts	Curb side pick up	Home For Good	Patrick Jane	DONE

Dashboard x FoodStrap x + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/past\_claim

## CLAIMS HISTORY

**Success Summary**

Number of People Fed

Congratulations! 310 people fed.  
Service to others is the rent you pay for your room here on earth.-- Mohammad Ali

Show 10 entries

Search: CLAIMED

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Volunteer	Status
10	Garlic Bread	Pizza	10	04/30/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Diary	None	Home For Good		CLAIMED

Showing 1 to 1 of 1 entries (filtered from 14 total entries)

Previous 1 Next

## FoodStrap Project Report

Dashboard x ec2-3-101-76-67.us-west-1.compute.amazonaws.com + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/shelter\_profile

SHELTER PROFILE

FOODSTRAP

HOME FOR GOOD



Phone  
8885554444

Email Id  
homeforgood@gmail.com

Address  
900

City  
San

State  
CA

Zip  
95112

UPDATE

Dashboard x DASHBOARD x + Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/volunteer\_dashboard

VOLUNTEER DASHBOARD

FOODSTRAP

HOW YOU CAN HELP?

PICK UP AND DELIVER FOOD



DATE



TIME



LOCATION

Show  
10 entries

Search:

#	Menu	Cuisine	Portions Count	Pick up Date Time	Address	Allergy Information	Special Instructions	Shelter	Status
1	Garlic Bread	Pizza	10	04/30/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Diary	None	Home For Good	CLAIMED

PICK UP

Showing 1 to 1 of 1 entries

Previous 1 Next

## FoodStrap Project Report

Dashboard    PICK UP

Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/pickup/5eab6b52f57fb7d4021e760

PICK UP

FOODSTRAP

Map

Garlic Bread

Cuisine Pizza Portion Count 10

Allergy Information

Dairy

Special Instructions None

Pick Up Time 04/30/2020 10:00 PM

Pick Up Address 935 N2nd Street, San Jose, CA 95112

Shelter Home For Good Status CLAIMED

CANCEL CONFIRM PICK UP

Dashboard    PROFILE

Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/vol\_profile

PROFILE

FOODSTRAP

PATRICK JANE

Phone 3004444555

Email Id pjane@gmail.com

Shelter Name

UPDATE

## 02. Localization

### 001. Hindi



#### अपना भोजन साझा करें

इमरो थोड़े कोई पीछावार को मौजूद पर लाने वाले में भी भोजन की इच्छा है। उन्हें किसानों द्वारा या बाजारों का सामान बांटने के बारे चर्चा नहीं करना पड़ता। अब भोजन दान करके मदद कर सकते हैं। आपके समर्पण की बहुत आवश्यकता है और बराहना की जाती है।

#### अधिशेष भोजन का दान करें

जाने के लिए सुरक्षित अधिशेष भोजन को काफ़ी करने के बाजार, स्थानीय भूमि गांव या गांवों को दान करें। यह एक दिल्ली तथा देश है, जहाँ पारिवारिक प्राकृति को रोकने के लिए लोड़ियां लगाने के बीच बदलना है, तो हम मदद कर सकते हैं।

#### भोजन ग्रहण करें

भोजनालय ये दान एवं लाभार्थी भोजन उन लोगों की सेवा कर सकते हैं जो भूमि का अभ्यन्तर करते हैं। यदि अपने किसान का भूमि लाभार्थी या स्थानीय का सामान खरीदने के बीच बदलना है, तो हम मदद कर सकते हैं।



#### हम क्या हैं?

हम साथ लोगों को अधिशेष भोजन के साथ साझीय प्रभुत्व लाहू एवं खिलाफों से जोड़कर अधिशेष पीढ़ियों भोजन फोड़ने का विकास प्रदान करते हैं। हम एक कुछ तंत्रार और रिपोर्टिंग नेटवर्क का नियंत्रित और रखरखाव करते हैं जो इन मौजूदा धर्मीय गांवों के माध्यम से उन लोगों के लिए भोजन के उपलब्ध भोजनों को जोड़ता है।

स्थानीय भूमि गांव या गांवों को दान करें। यह एक दिल्ली तथा देश है, जहाँ पारिवारिक प्राकृति को रोकने के लिए लोड़ियां लगाने के बीच बदलना है, तो हम मदद कर सकते हैं।

स्थानीय समुदायों को साथ देने हैं।



#### हम क्या हैं?

हम ऐसे लोगों की मदद करते हैं जो अधिशेष भोजन के साथ साझीय प्रभुत्व लाहू एवं खिलाफों से जोड़कर अपनी नीतियों की दी है, या कार्य मोड़िफिकेशन इमर्जेंसी है, या उनकी काफ़ दूर नहीं है। हम उन लोगों की मदद करते हैं जो जलवायन के द्वारा दूषित हो गए हैं और उन्हें भोजन पाने में कुछ मदद की ज़रूरत है। उन साथ तक उन्हें रहने वाले लोगों की ज़रूरत है, जो एक अपार्टमेंट के रूप में बदल जाएंगे। युद्धों में लात भी अधिशेष लोगों के बीच बदल जाता है, तो हम मदद कर सकते हैं।

#### जहाँ से हम अपना भोजन मिलता है?

हम एसे लोगों की मदद करते हैं जो अधिशेष भोजन के साथ साझीय प्रभुत्व लाहू एवं खिलाफों से जोड़कर अधिशेष पीढ़ियों भोजन फोड़ने का विकास प्रदान करते हैं। हम एक कुछ तंत्रार और रिपोर्टिंग नेटवर्क का नियंत्रित और रखरखाव करते हैं जो इन मौजूदा धर्मीय गांवों के माध्यम से उन लोगों के लिए भोजन के उपलब्ध भोजनों को जोड़ता है।

#### जहाँ से हम अपना भोजन मिलता है?

हम एसे लोगों की मदद करते हैं जो अधिशेष भोजन के साथ साझीय प्रभुत्व लाहू एवं खिलाफों से जोड़कर अपनी नीतियों की दी है, या कार्य मोड़िफिकेशन इमर्जेंसी है, या उनकी काफ़ दूर नहीं है। हम उन लोगों की मदद करते हैं जो जलवायन के द्वारा दूषित हो गए हैं और उन्हें भोजन पाने में कुछ मदद की ज़रूरत है। उन साथ तक उन्हें रहने वाले लोगों की ज़रूरत है, जो एक अपार्टमेंट के रूप में बदल जाएंगे। युद्धों में लात भी अधिशेष लोगों के बीच बदल जाता है, तो हम मदद कर सकते हैं।

Dashboard x Foodstrap - समस्या

Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/problem

FOODSTRAP भर समस्या भोजन दान भोजन प्राप्त स्वयंसेवक

साइन इन परिस्थिति

भोजन की असुरक्षा

वर्तमान में, दुनिया में लाभी को लिहाजे के लिए पर्याप्त से अधिक भोजन का उत्पादन होता है, फिर भी 815 मिलियन लोग (ऐक्षिक आबादी का समाप्त 11%) 2016 में भूख से मर गए, मृत्यु, अस्थिरी की कृषि विभाग (पूर्वोत्तर) एक लॉकिं, स्वस्थ जीवन के लिए पर्याप्त भोजन की तागाहार पहुँच की कमी के रूप में लाभ अद्युक्तों की परिमापित करता है।

खाना बर्बाद.

भोजन दान करना आसान है

हर दान से फर्क पड़ता है

दयालुता का सबसे सरल कार्य प्रार्थना में छुटकने वाले एक हजार से अधिक शक्तिशाली हैं।

— मानव सभा

इनमें से कई कर्मी विदेशी मदद की ज़रूरत है। उन्हें किराया देने या स्वस्थ किराये का सामान खरीदने के बीच बदन नहीं करना चाहिए। अप भोजन दान करके मदद कर सकते हैं। आपके समर्पण की भूमि आपको कहा है और समझना की ज़रूरत है।

स्वरूप के अन्तर्गत, हमारे भोजन की अपीली का 30-40% के बीच भोजन की बढ़ती का अनुमान है, प्रायः लगभग 72 विभिन्न पार्टी द्वारा जर्वे भोजन होता है।

अनन्तों के ब्रेंड, देशों की दौड़ के बीच, और एक लंदर में उत्पादन के, दर्द स्थानीय गैर-लाभकारी संस्करणों और उन सोनों के खिलाफ़ करें।

सिविल-ताक़िल, राष्ट्रीय भोजनात्मक शुद्धता से सेकर दुनिया भर में मेघा कांडी शायद, जो अमेरिका के हर सड़क के कोने में फैला हुआ है, इन प्रतिक्रियाओं की जीव वर्णों द्वारा दिन के अंत में दान करके चारों को जानकार के रिपोर्टरों द्वारा छोड़ दिया जाता है।

**Dashboard** **भोजन प्राप्त करें** **Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/shelters**

**FOODSTRAP** घर समस्या भोजन दान भोजन प्राप्त स्वयंसेवक साइट नियम

### भोजन सहायता प्राप्त करें

दायरुता का सबसे सरल कार्य प्रार्थना में दूकाने वाले एक हजार से अधिक शक्तिशाली हैं।  
— रामेश गुप्ता

पिछले चार, बंगलुरु राज्य अमेरिका में लगभग 4.1 मिलियन टन भोजन उत्पन्न किया गया है। जबकि अमेरिकी लाली टन भोजन का निपटान करते हैं, अमेरिकी कृषि विभाग का अनुमान है कि 11.0 प्रतिशत अमेरिकी पर्दे - लगभग 15 मिलियन घरों में - संसाधनों की कमी के कारण अपने लभी लदर्दी के लिए चर्चित भोजन प्राप्त करने में कठिनाई हुई। कई मामलों में, हमारे देश के लैक्षणिक में केक दिया गया भोजन पोषण, लालू और लोडन है।

यदि आपको अपने आधय के लिए भोजन सहायता चाही आवश्यकता है, तो अभी साइट अप करें।

**अभी साइट अप करें!**

**स्वयंसेवक के लिए यह हमेशा एक शानदार अवसर होता है।**

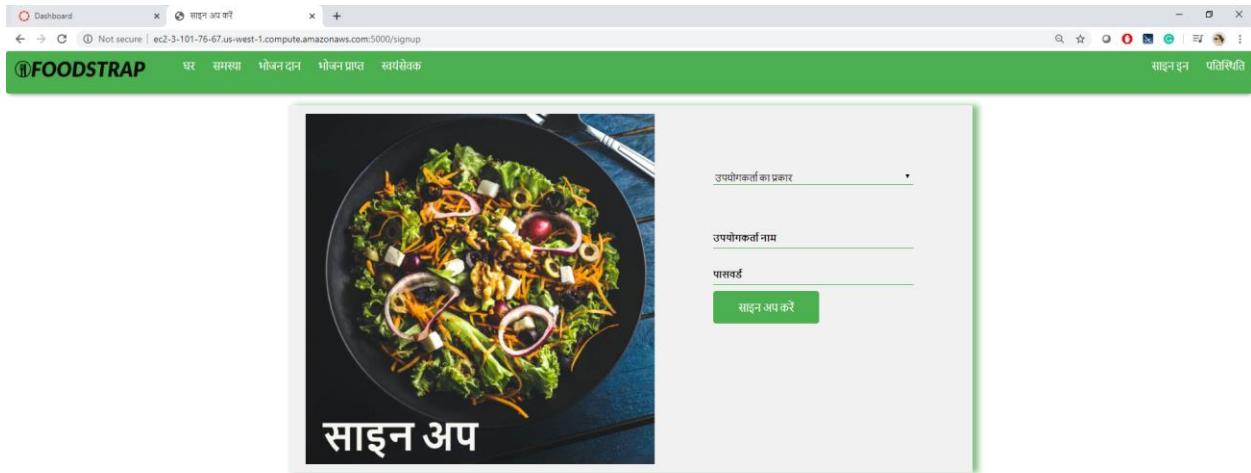
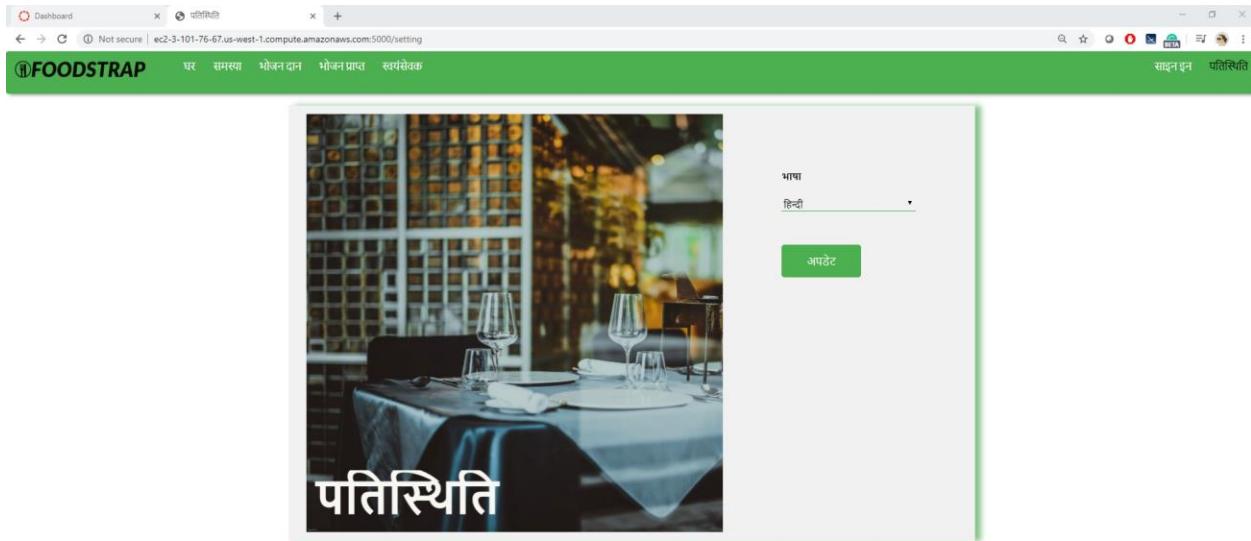
बचाव के लिए स्वयंसेवक ड्राइवर  
स्वयंसेवक फूट-फुटर्स के द्वारा और आप हैं ज्यादा के द्वारा रेसर्व्स और आश्रय के बीच पूल हैं। विभिन्न रेसर्व्स से जाश्नों को अधिकोष भोजन के परिवहन का समर्पन करने के लिए, स्वयंसेवक लालूक तरही में जाते हैं। एक स्वयंसेवक लालूक के रूप में, आप इस राज्यान्वयन का हिस्सा होंगे, जो लालू असुरक्षा की समस्या की हुत करने में एक उत्ताह है।

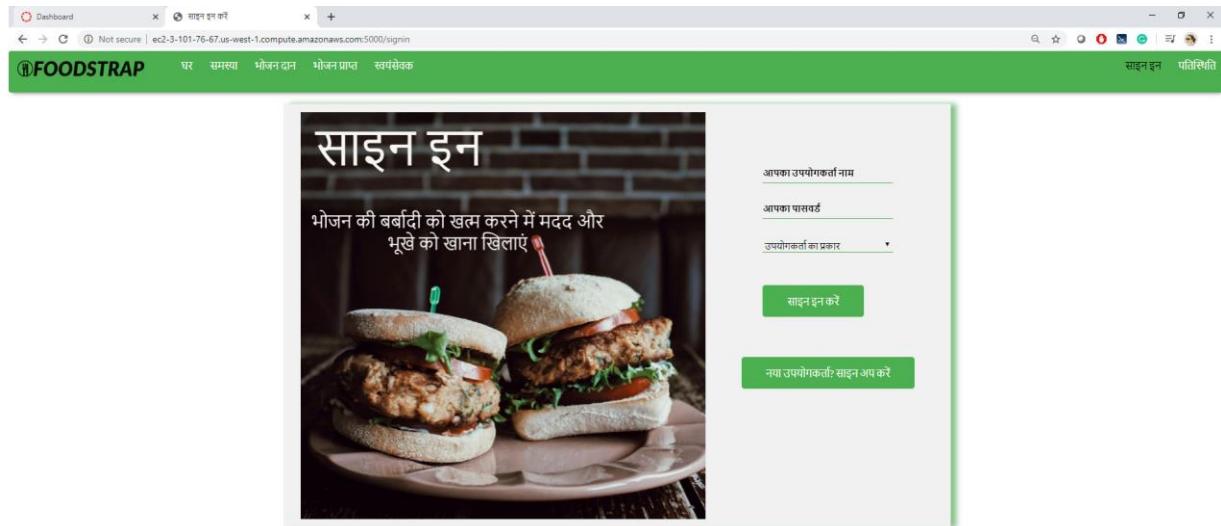
रुचि है? साइट अप करने से चढ़ते एक स्टॉप येक.

इसमें लिखा द्राइवर के रूप में लालू सेवा करने से यहते, हम कुछ अदायकतामों या योग्यताओं को तुम्हीनकद करना चाहते हो एक सफल स्वयंसेवक होने की आवश्यकता हो सकती है।

- 1. लालूक लालूसे और बीमा,
- 2. लुट्र का कार्टीफोन,
- 3. बाहन का उपयोग,
- 4. एलिकेमन का उपयोग करने की क्षमता,

क्या आप मदद करने के इच्छुक हैं?  
**अभी पंक्तीकरण करें!**





Dashboard      Not secure | ec2-3-101-76-67.us-west-1.compute.amazonaws.com:5000/signin

भोजनात्मक प्रबंधक डैशबोर्ड

FOODSTRAP

आप कैसे मदद कर सकते हैं?  
अधिक भोजन दान करें

तारीख स्थान नोट  
समय भोजन संख्या भोजन

पूर्ण वेस्टर्न एसोसिएशन के अनुसार, अमेरिका का एक एकल रेतारा साल में लगभग 100,000 पार्ट्स भोजन बर्बाद करता है, जिससे उन्हें धूम से रोते होने वाले स्मार्टफोन के लिए शाम दान मिल जाता है। ऐसा सिविल और आपराधिक दृष्टिकोण सुनिश्चित है, एक प्रायोगिकों की दृष्टि की गयी भोजन के परिवर्तन स्वरूप बीमा या शोटेल सेवा यारियां भोजन की बर्बादी पर एक राष्ट्रीय ध्वनि कानून के बावजूद, अमेरिकी रेतारा अभी भी अपने अपीलिंग भोजन का 94 प्रतिशत लोका देते हैं।

आज दान करें और जीवन बदलें

127

**भोजनालय प्रबंधक टैशबोर्ड**

आप कैसे मदद कर सकते हैं?  
अधिक भोजन दान करें

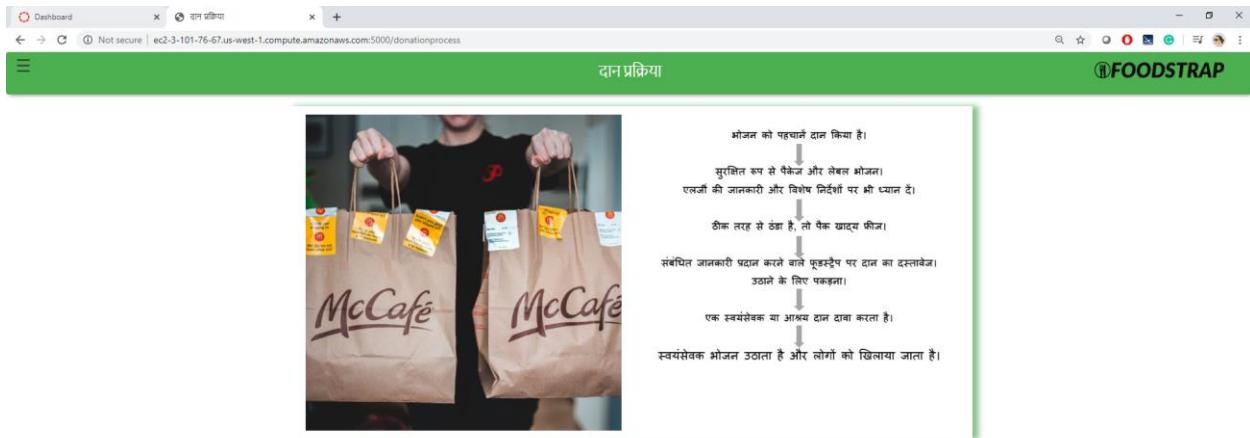
तारीख स्थान नोट  
समय भोजन संछेत्र भोजन

ग्रीन रेस्टोरेंट्सियल के अनुसार, अमेरिका का एक लाख से ज्यादा साल में लाभाम 100,000 उड़ान भोजन बर्चेट बनता है, जिससे उन्हें भूल से गाते दें और उसके के लिए भूल दान मिल जाता है। इसके लिए और आरोपित भूल दान के लिए है। एक प्राप्ति वित्ती को दान किए गए भोजन के प्रतिशत का या सांकेतिक होना चाहिए। भोजन की वर्षीय पर एक राष्ट्रीय धान कॉटिंग के साथ दूर, अमेरिकी रेस्टोरेंट्सी में अनेक जीवन भोजन का 94 प्रतिशत बीमा देते हैं।

आव दान करें और जीवन बचाएं।

**दान**

भोजनसूची  
भोजन के प्रकार भोजन संख्या  
एर्की दी जानकारी  
दिनें निर्देश  
समय लेने  
पता  
शहर प्रदेश जिला  
दान



आवश्यक खाद्य सहायता? भोजनालय से दान स्वीकार करें

ज़रूरत हिलाऊ

भोजन राशिए सुखा है। भोजन अपील्यवस्था है। यह रोजाना, उच्च, इतिहास है। लोग नी रख कुछ हैं। - जैस एडेंस

खाद्य असुधा बह रही है और भूल इके साथ निकटता से संबंधित है। इस बढ़ती हुई समस्या का समाधान करने के लिए, बेपर आश्रय, लाल बैक्स और सुन रोइ जैसे संस्थान ने मुझे को भोजन देकर अपना काम कर रहे हैं, स्वयंसेवकों की सहायता से।

हमारा स्थान  
900 N1st Street, San Jose, CA - 95112

**शेटर डैशबोर्ड**

आवश्यक खाद्य सहायता? भोजनालय से दान स्वीकार करें




आपको कब चाहिए



आपको कितने की आवश्यकता है

**ज़रूरत छिलाओं**  
भोजन रणनीति सुनाता है। भोजन अधिकारीया है। यह रोकगार, जल्दी, इतिहास है। कौन ही चाहे कुछ है। - जॉस फ्रेंस  
खाद्य अनुदान बढ़ रही है और भूत दूसरे साथ निकटता से समीक्षा है। इस बढ़ती हुई समस्या का समाधान करने के लिए, बैपर आयोग, लाल कैंडी और सूर्य रसाई जैसे संस्थान भूमि की भोजन देनार अपना काम कर रहे हैं, खपेसेकों की सहायता से।

हमारा स्थान  
900 N1st Street, San Jose, CA - 95112

**हमारे स्थानोंका**

स्वीकृति की पुष्टि करें

**पोशाकी**

पोशाक के प्रकार	प्रोत्तम संख्या
Italian	30
दूसरी की जगहकी	Nuts, diary
प्रियंका निर्देश	None
माप देने	04/30/2020 10:00 PM
स्थान	935 N2nd Street, San Jose, CA 95112
सिलि	OPEN
रद्द करना	<b>स्वीकार करें</b>

The screenshot shows the FoodStrap volunteer dashboard. At the top, there are three green icons: a hand holding a leaf, a fork and knife, and another hand holding a leaf. Below them, the text reads "आप कैसे मदद कर सकते हैं? उठाओ और भोजन वितरित करें". To the right of the icons are three circular icons: a calendar labeled "तारीख", a clock labeled "समय", and a location pin labeled "स्थान". A table below shows one entry:

#	भोजनसुधी	भोजन के प्रकार	भोजन संख्या	समय लेने	पता	एकार्डी की जानकारी	विशेष निर्देश	आवश्यक	स्थिति
1	Garlic Bread	Pizza	10	04/30/2020 10:00 PM	935 N2nd Street, San Jose, CA 95112	Diary	None	Home For Good	CLAIMED

At the bottom left, it says "Showing 1 to 1 of 1 entries". On the right, there are "Previous" and "Next" buttons. A green button at the bottom right says "उठाओ".

The screenshot shows the FoodStrap volunteer detail view for a donation. On the left, there is a sidebar with navigation links: "देशबोर्ड", "शिक्षियत", and "लॉग आउट". The main area has a green header "उठाओ". It features a map of the San Francisco Bay Area with a red marker indicating the pickup location. To the right of the map is a detailed form for the donation:

भोजनसुधी Garlic Bread	भोजन के प्रकार Pizza
भोजन संख्या 10	
एकार्डी की जानकारी Diary	विशेष निर्देश None
समय लेने 04/30/2020 10:00 PM	
पता 935 N2nd Street, San Jose, CA 95112	
आवश्यक Home For Good	स्थिति CLAIMED

A green button at the bottom right says "उठाओ".

## 002. French

**FOODSTRAP**

Si vous ne pouvez pas nourrir une centaine de personnes, n'en nourrissez qu'une!  
— Mère Teresa

**Partagez votre nourriture**

De nombreuses familles de notre région ont besoin d'un peu d'aide pour mettre de la nourriture sur la table. Ils ne devraient pas avoir à choisir entre payer un loyer ou acheter une épicerie salée. Vous pouvez aider en faisant des dons de nourriture. Votre soutien est grandement nécessaire et apprécié.

**Donnez un surplus de nourriture**

Au lieu de jeter les surplus de nourriture qui sont sûrs à manger, faites un don aux organisations locales de lutte contre la faim. Cela offre un triple avantage: la nourriture reste hors des décharges pour éviter l'impact environnemental, les organisations à but non lucratif bénéficient des dons et les restaurants redonnent aux communautés locales.

**Recevoir de la nourriture**

La nourriture salée et de qualité préparée donnée par les restaurants peut servir ceux qui ont faim. Si vous devez choisir entre payer un loyer ou acheter une épicerie salée, nous pouvons vous aider.

## LE PROBLÈME

**Insécurité alimentaire**

wages  
**Food insecurity** vs problem  
lack resources  
financial poverty  
people needs

Le monde produit actuellement plus qu'assez de nourriture pour nourrir tout le monde, mais 815 millions de personnes (environ 11% de la population mondiale) ont souffert de la faim en 2016, selon l'UN. Le département américain de l'Agriculture (USDA) définit l'insécurité alimentaire comme un manque d'accès constant à suffisamment de nourriture pour une vie active et salée.

Déchets alimentaires.

The screenshot shows a web browser window for the FoodStrap website. The title bar indicates it's a 'Not secure' connection. The main content area features a large green banner with white text: 'C'EST TOUJOURS UNE MERVEILLEUSE OCCASION DE FAIRE DU BÉNÉVOLAT.' Below the banner is a photo of three smiling volunteers in blue shirts taking a selfie. The text 'Des chauffeurs bénévoles à la rescousse' is followed by a paragraph about the role of volunteers in transporting surplus food from restaurants to refugees. A list of requirements for drivers is provided, along with a 'S'inscrire maintenant' button.

**C'EST TOUJOURS UNE MERVEILLEUSE OCCASION DE FAIRE DU BÉNÉVOLAT.**

**Des chauffeurs bénévoles à la rescousse**

Les bénévoles sont le cœur et l'âme de FoodStrap car ils sont le pont entre les restaurants donneurs et les refuges. Pour soutenir le transport des surplus de nourriture de divers restaurants aux refuges, des chauffeurs bénévoles entrent en scène. En tant que chauffeur bénévole, vous ferez partie de ce projet passionnant qui tente de résoudre le problème de l'insécurité alimentaire.

Intéressé ? Un contrôle unique avant de vous inscrire.

Avant de faire du bénévolat en tant que chauffeur pour nous, nous aimerions énumérer certaines exigences ou qualifications dont on pourrait avoir besoin pour être un bénévole réussi.

1. Permis de conduire et assurance.
2. Posséder un smartphone
3. accès véhicule.
4. Possibilité d'utiliser des applications.

Es tu intéressé?

S'inscrire maintenant