|  |  |  |  |
| --- | --- | --- | --- |
| **Problem Definition** | **FOOD WASTE MANAGEMENT** |  | |
| **Student Name** | **PAYAL KHAH** | |  |
| **Student Enrolment Numbers** | 190110107080 | |  |

**Food Waste Management**

There is growing evidence that a significant share of global food is thrown away. Reducing food waste is a key sustainability challenge for the food service industry.Everyday people are wasting food. We are designing this website in order to reduce that food wastage problem. In this software the donor(the one who has extra food) can enter their Location, amount of food they want to distribute.Then a simple notification is generated in the website and the agent among that location can login & can gather the details of the food donor.The donor can hold an account in this application & whenever there is food wastage he can login and enter the details of food and location. The agent can also hold an account and can retrieve the details. After retrieving the details the agent can collect food from the donor and can redistribute to the orphans or others.

**Functional requirements** are description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called **Functional Specification**.

In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behaviour of the system.

* **Functional Requirements**

1. NGOs
   1. Authentication
   2. Uploading organization info.
   3. Booking acceptance/decline
   4. Room for cancellation
   5. Customer Feedback
2. Users
   1. Location based service
   2. User information
   3. Usage history
   4. User authentication
   5. Addition and Modification of information
   6. Assign user passwords
   7. Be a Volunteers

**Non-functional Requirements** specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

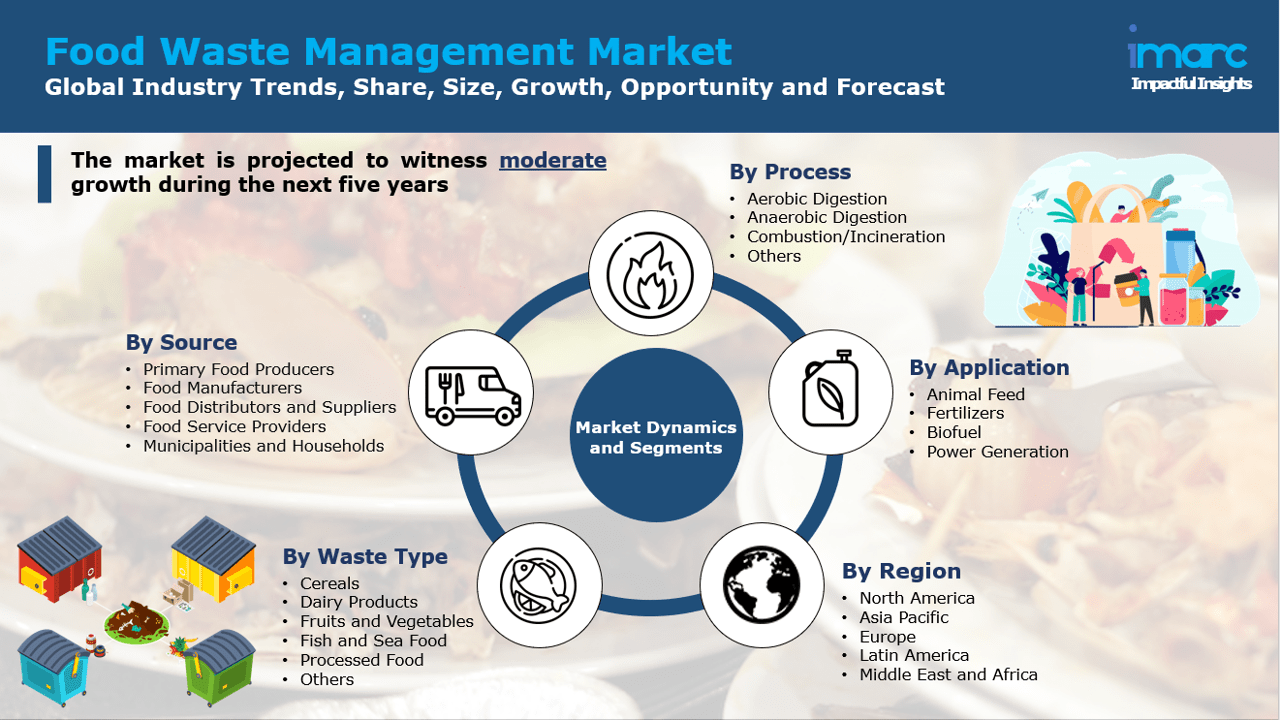
Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

* **Non-Functional Requirements**

1. Interactive UI
2. Database Security and availability
3. Good search module
4. News Feed
5. Availability of Volunteers
6. Mobile friendly
7. Good response time
8. Live Tracking
9. Better recommendations

**MARKET SURVEY**

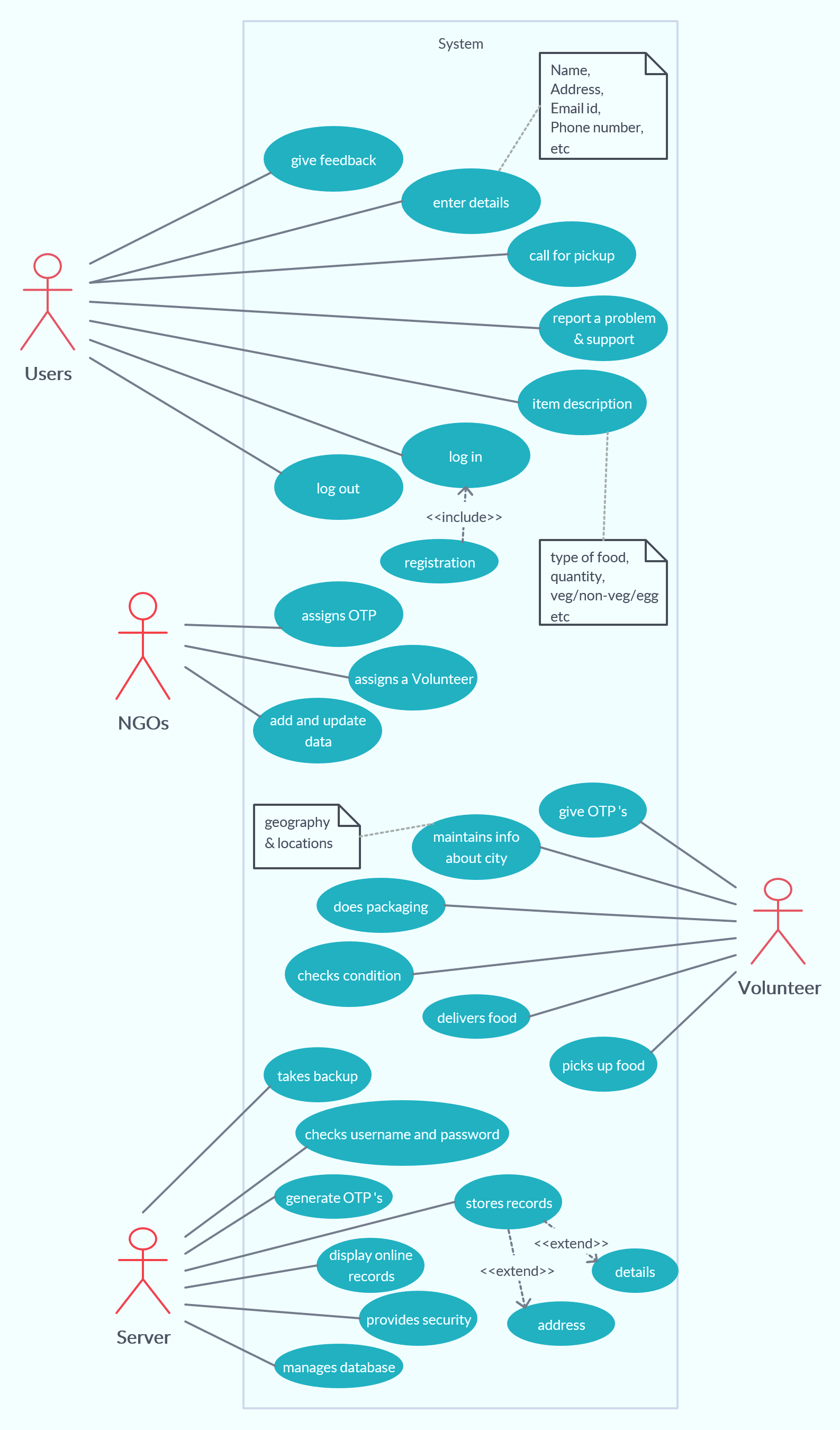
The global food waste management market grew at a CAGR of around 5% during 2015-2020. Food waste management involves various activities that are essential for disposing of food waste. It also includes the waste prevention, collection, treatment, recovery, transportation as well as recycling with the help of various management procedures. Some of these disposal techniques include landfills, composting, and anaerobic digestion of the generated wastes that are being employed across the commercial and residential segments. The increasing requirement for food waste management can be largely attributed to the rise in food processing waste, which includes agricultural, dairy food, poultry and seafood processing waste.



**USE CASE**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

An **entity–relationship model** describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between [entities](https://en.wiktionary.org/wiki/entity) (instances of those entity types). In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. The ER model becomes an abstract [data model](https://en.wikipedia.org/wiki/Data_modeling), that defines a data or information structure which can be implemented in a [database](https://en.wikipedia.org/wiki/Database), typically a [relational database](https://en.wikipedia.org/wiki/Relational_database).

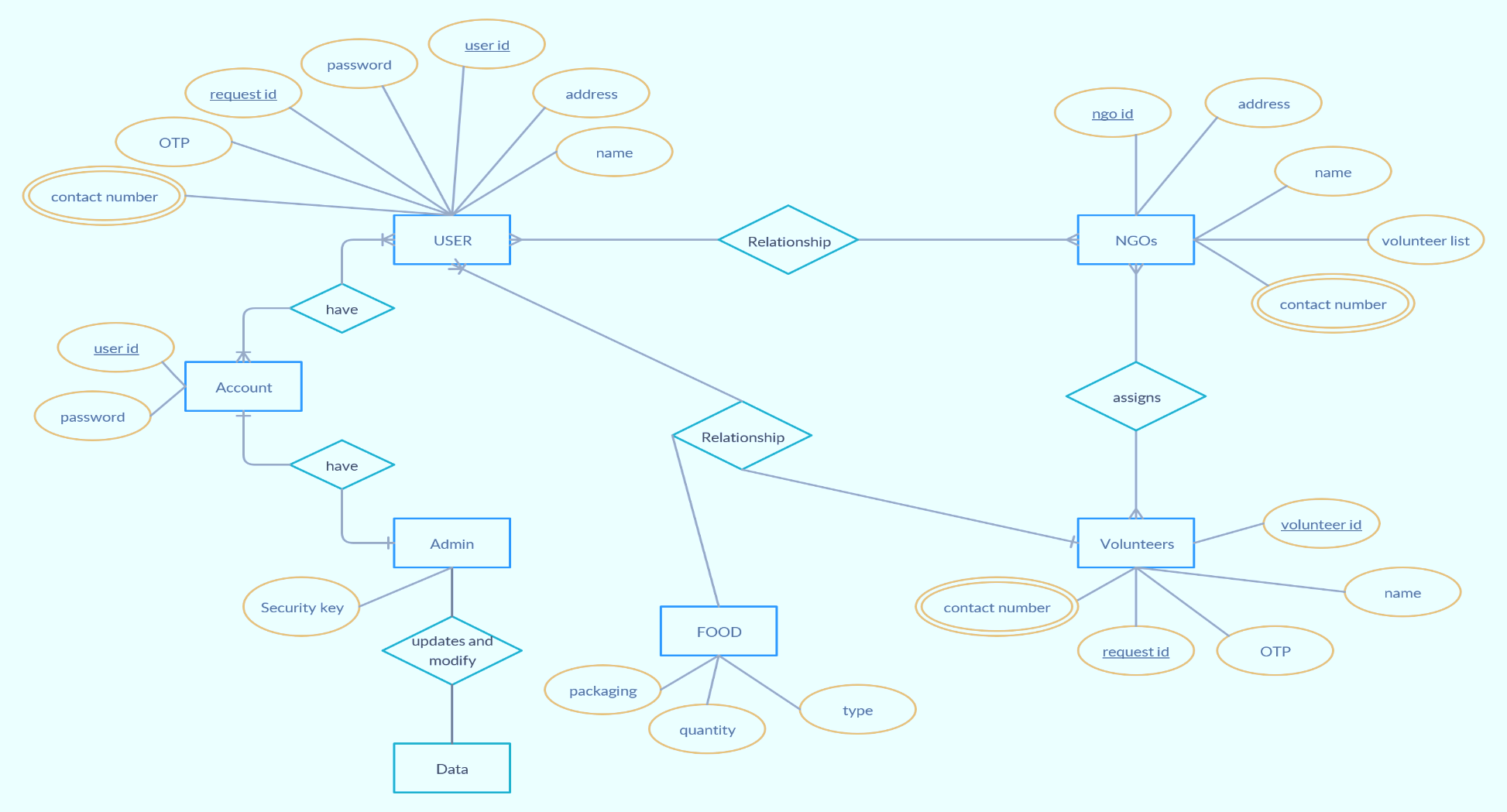


**USE CASE**

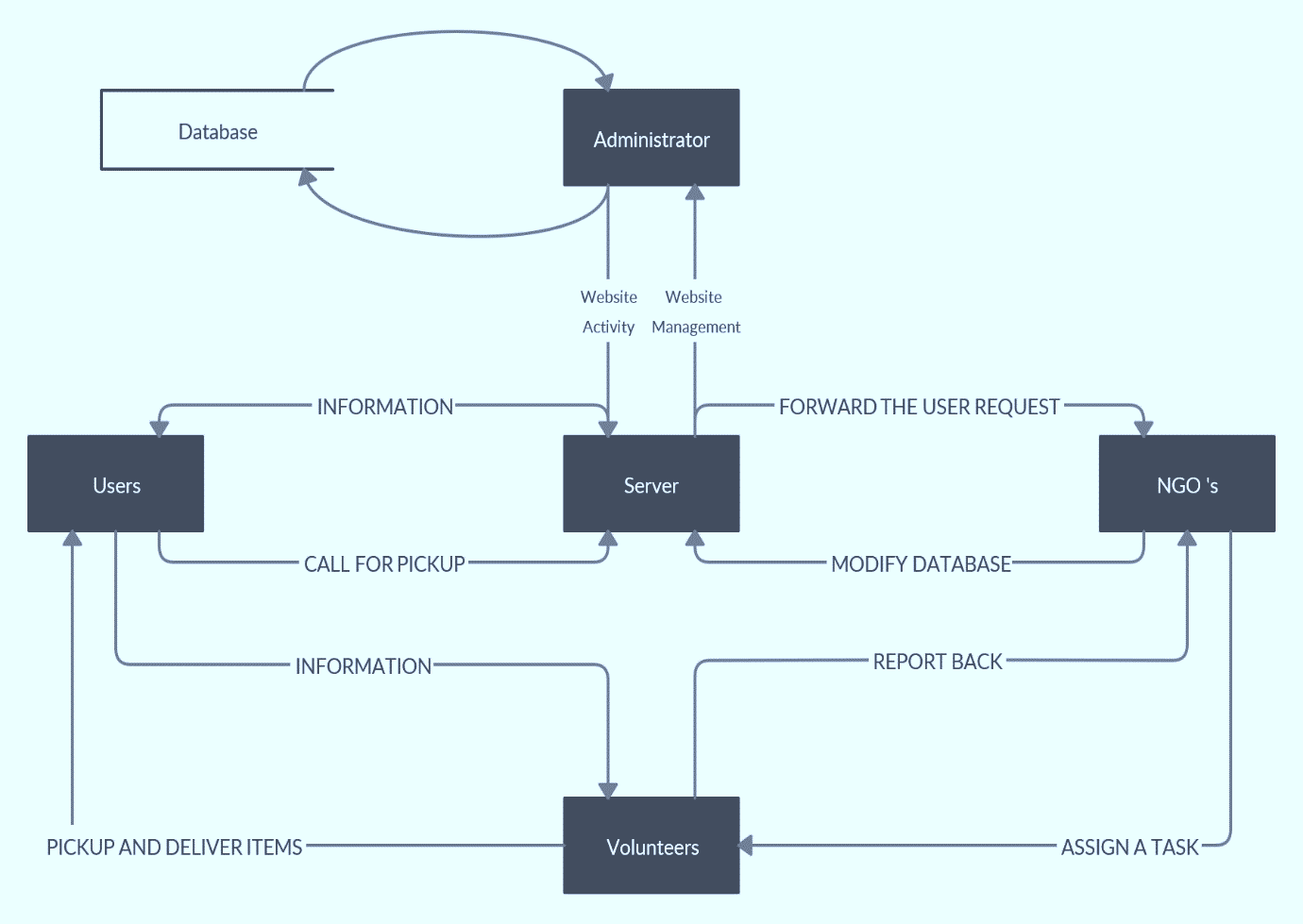
**DFD**

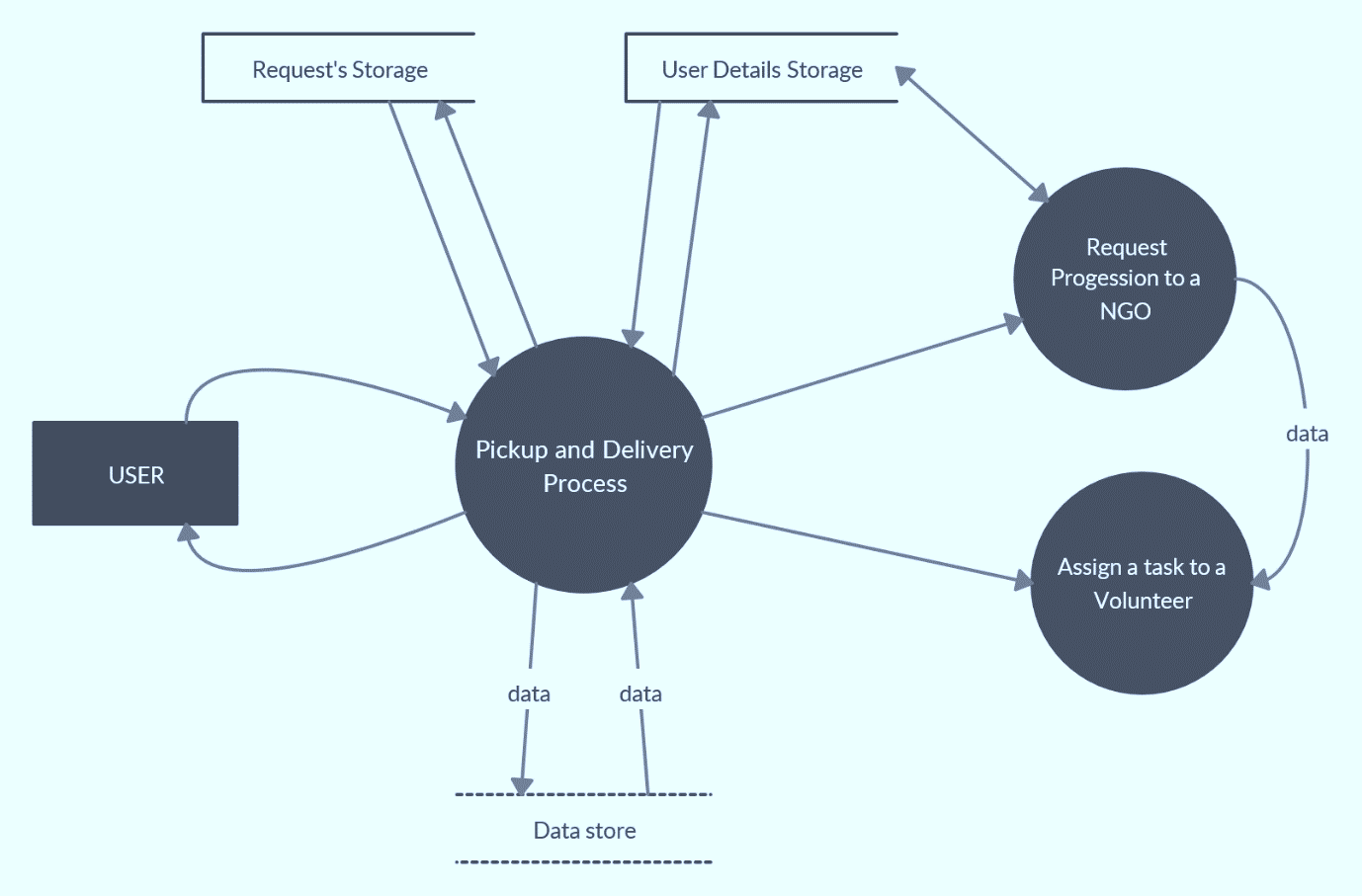
A data-flow diagram (DFD) is a way of representing a flow of a data of a [process](https://en.wikipedia.org/wiki/Process) or a system (usually an [information system](https://en.wikipedia.org/wiki/Information_system)). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a [flowchart](https://en.wikipedia.org/wiki/Flowchart).

A **data dictionary** is a file or a set of files that includes a database's metadata. The data dictionary holds records about other objects in the database, such as data ownership, data relationships to other objects, and other data. The data dictionary is an essential component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

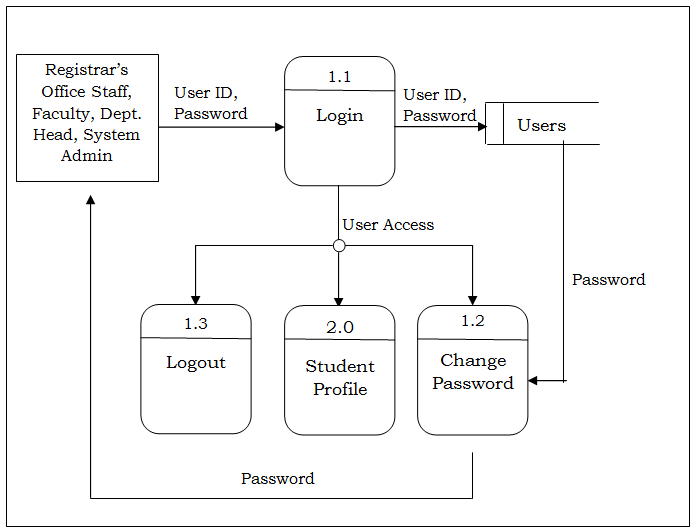


**DFD : LEVEL 0**





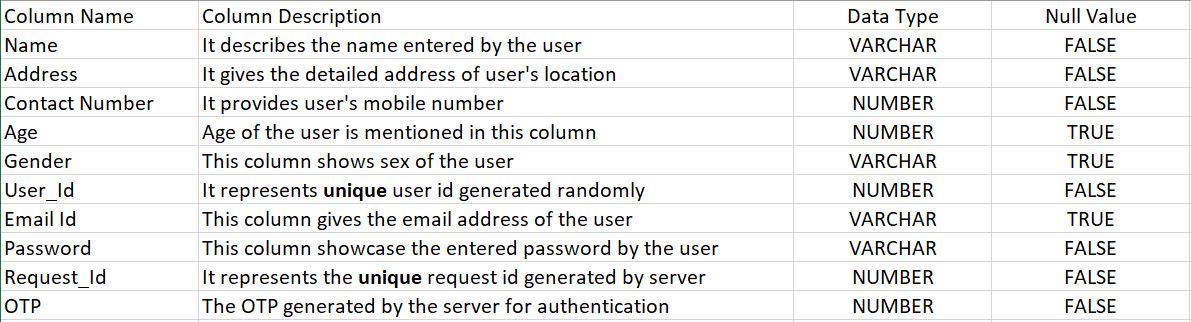
**LEVEL - 1 DFD**

****

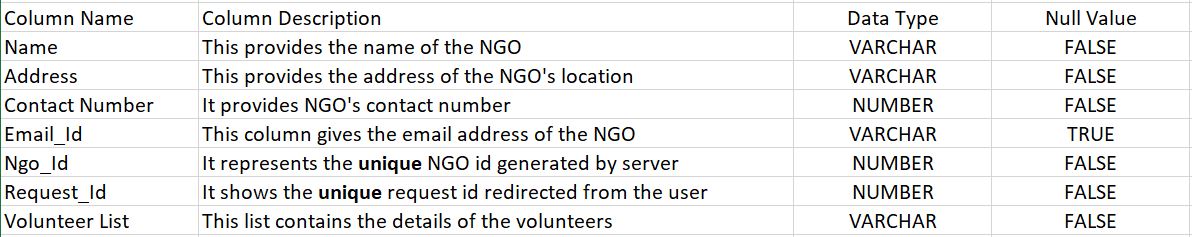
**LEVEL – 2 DFD**

**DATA DICTIONARY**

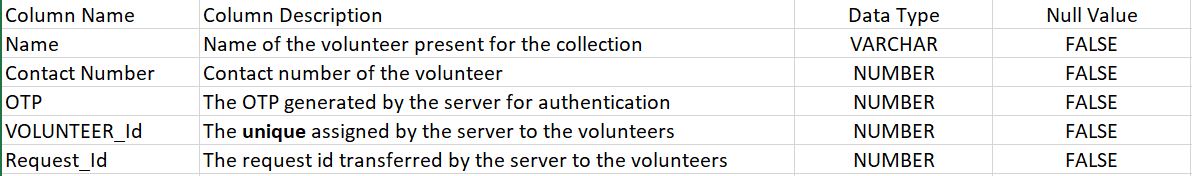
**Table: - User**



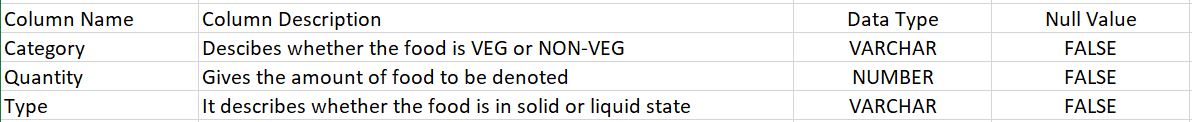
**Table: - NGO**

****

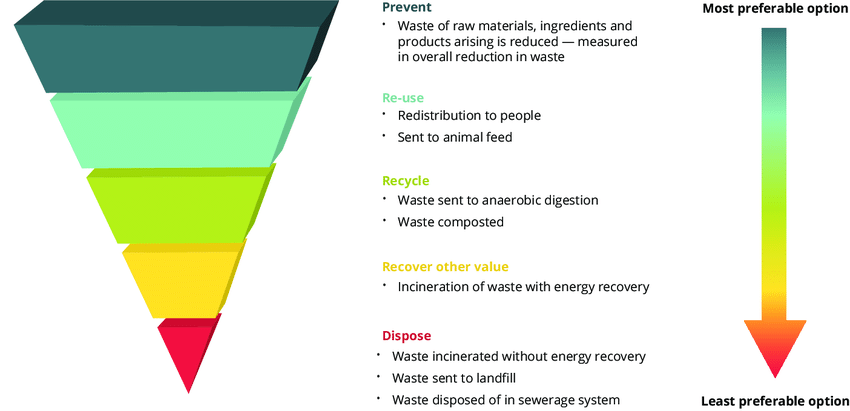
**Table: - Volunteer**

****

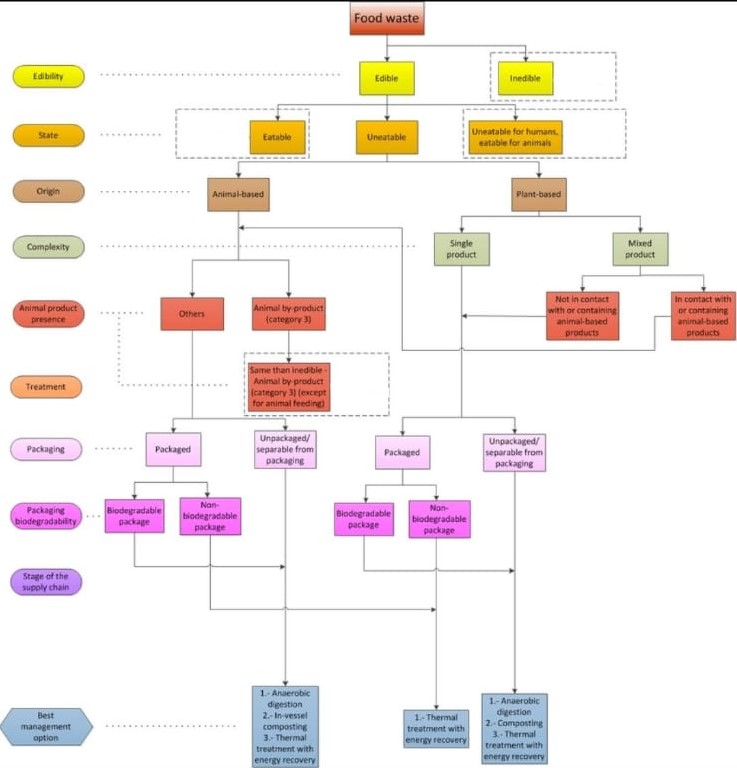
**Table: - Food**

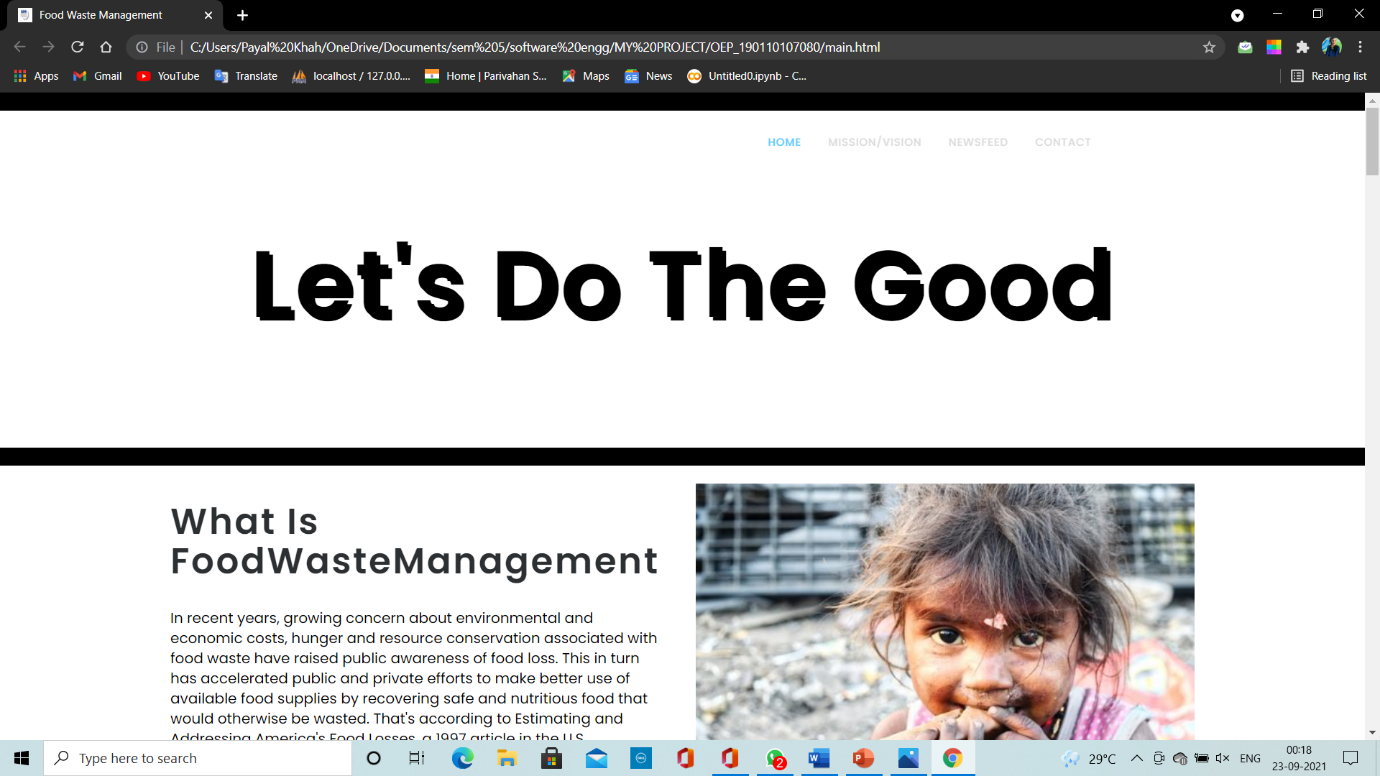
****

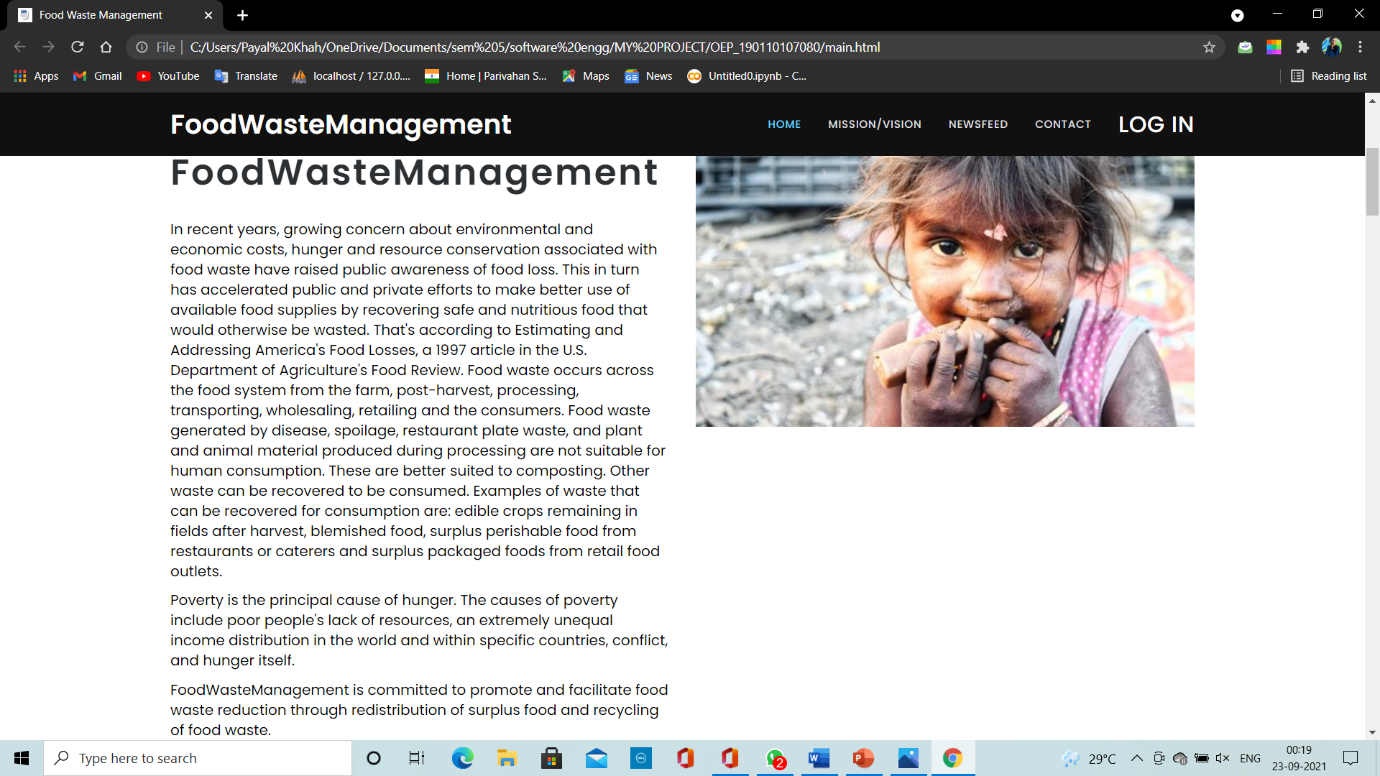
**ACTIVITY DIAGRAM**

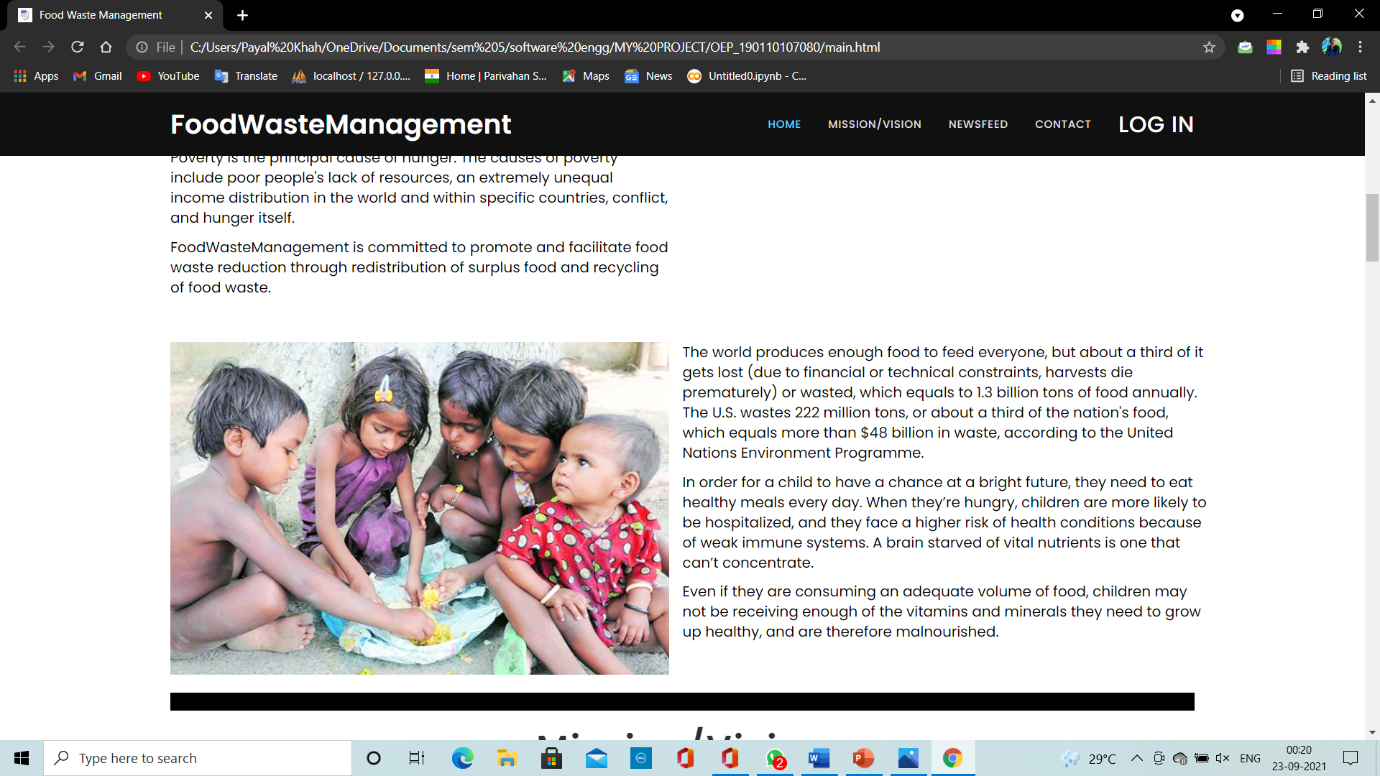
****

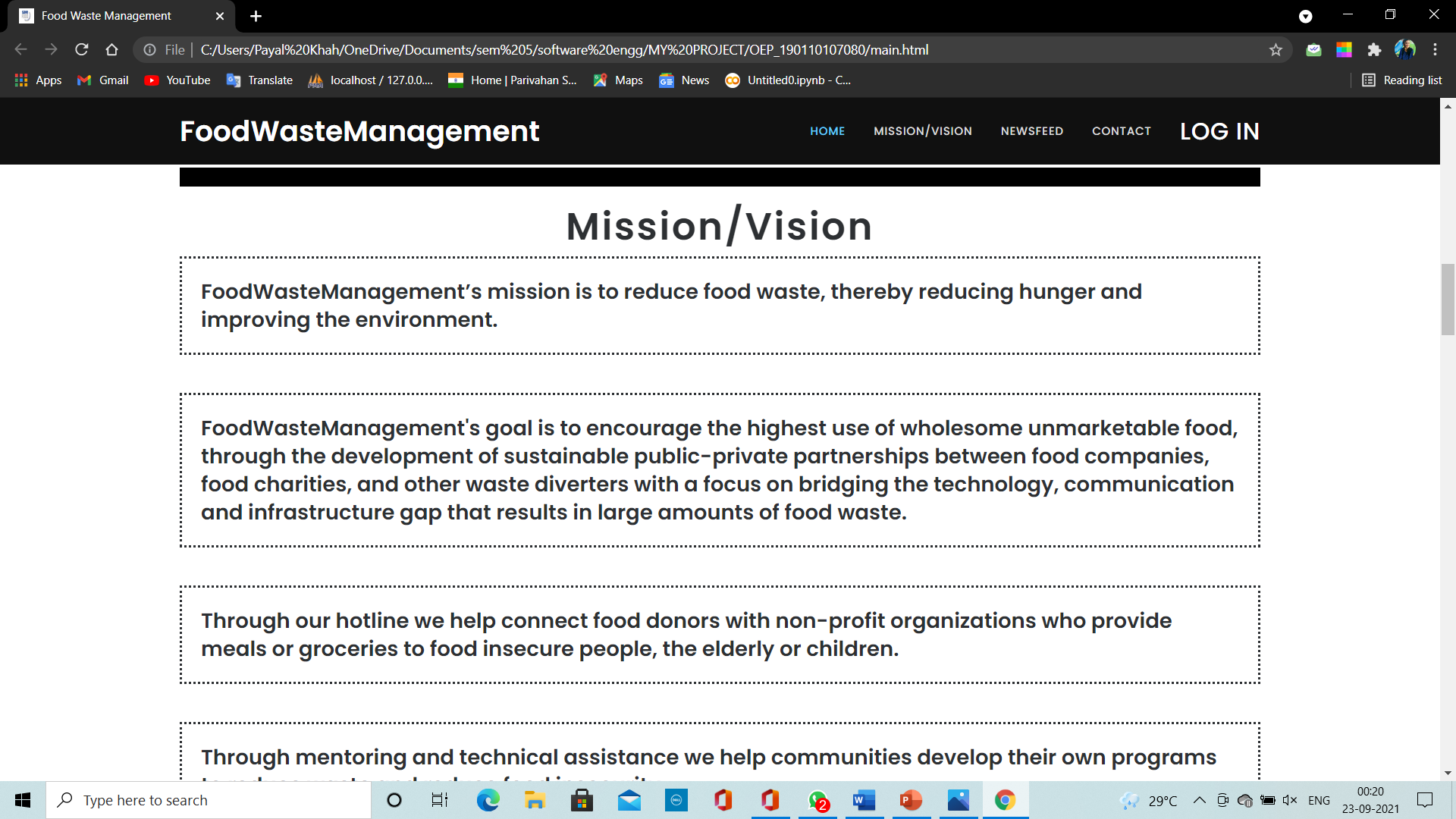
**CLASS DIAGRAM**

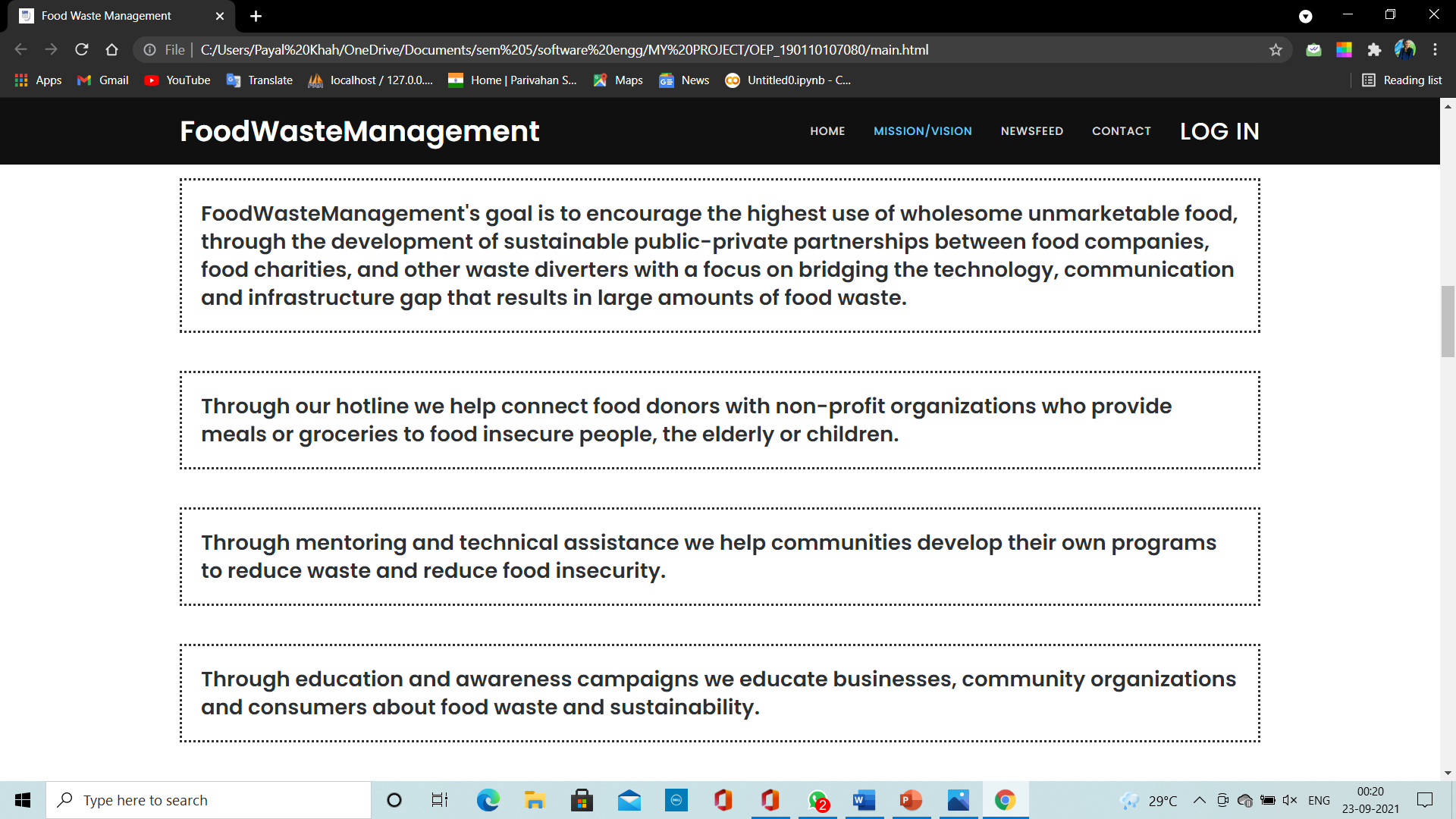


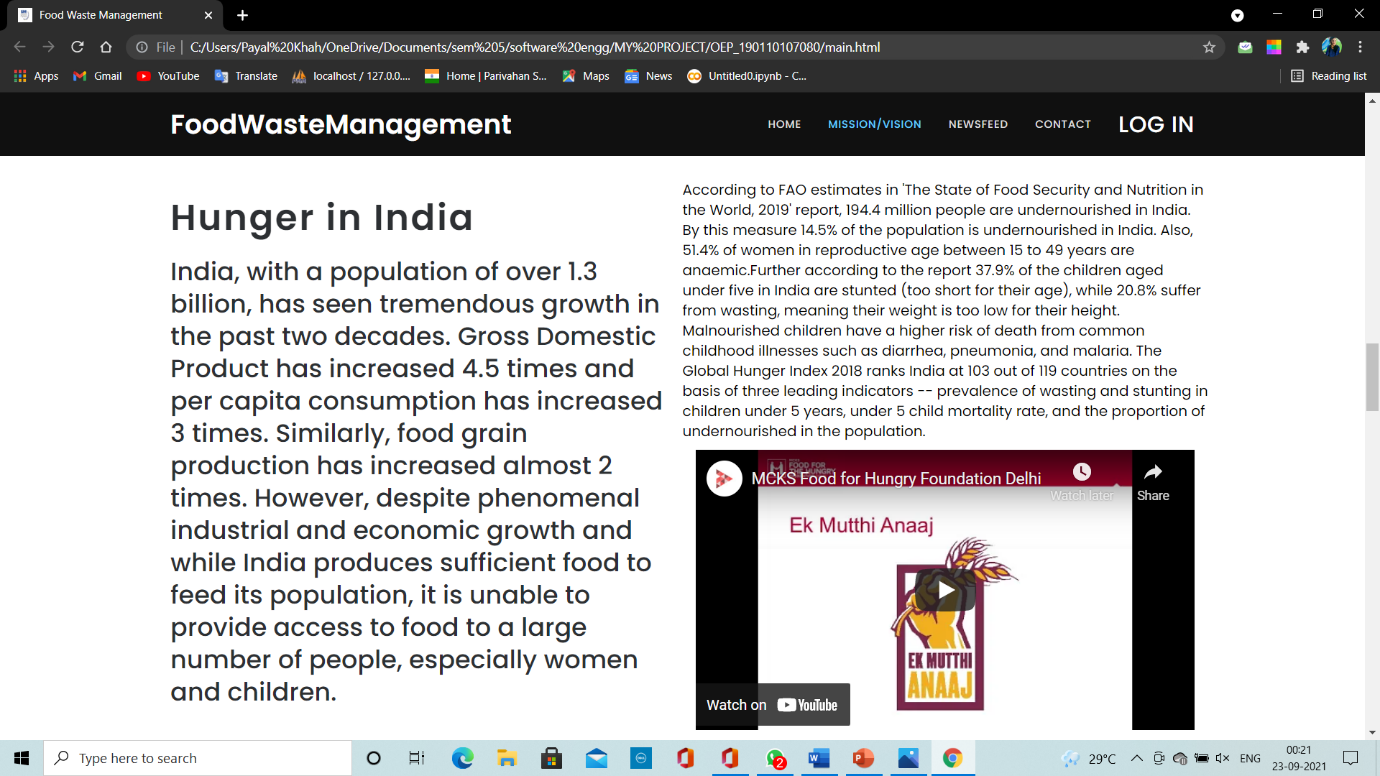


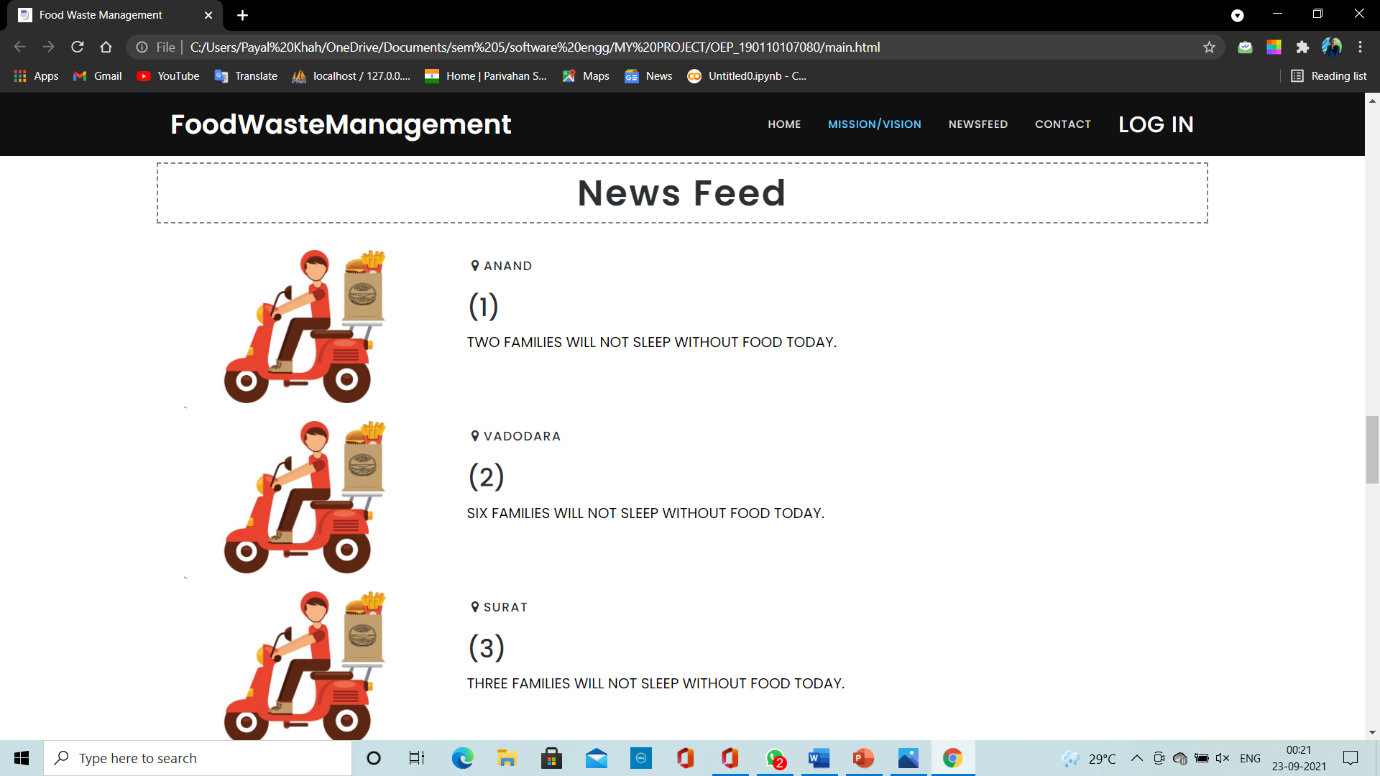


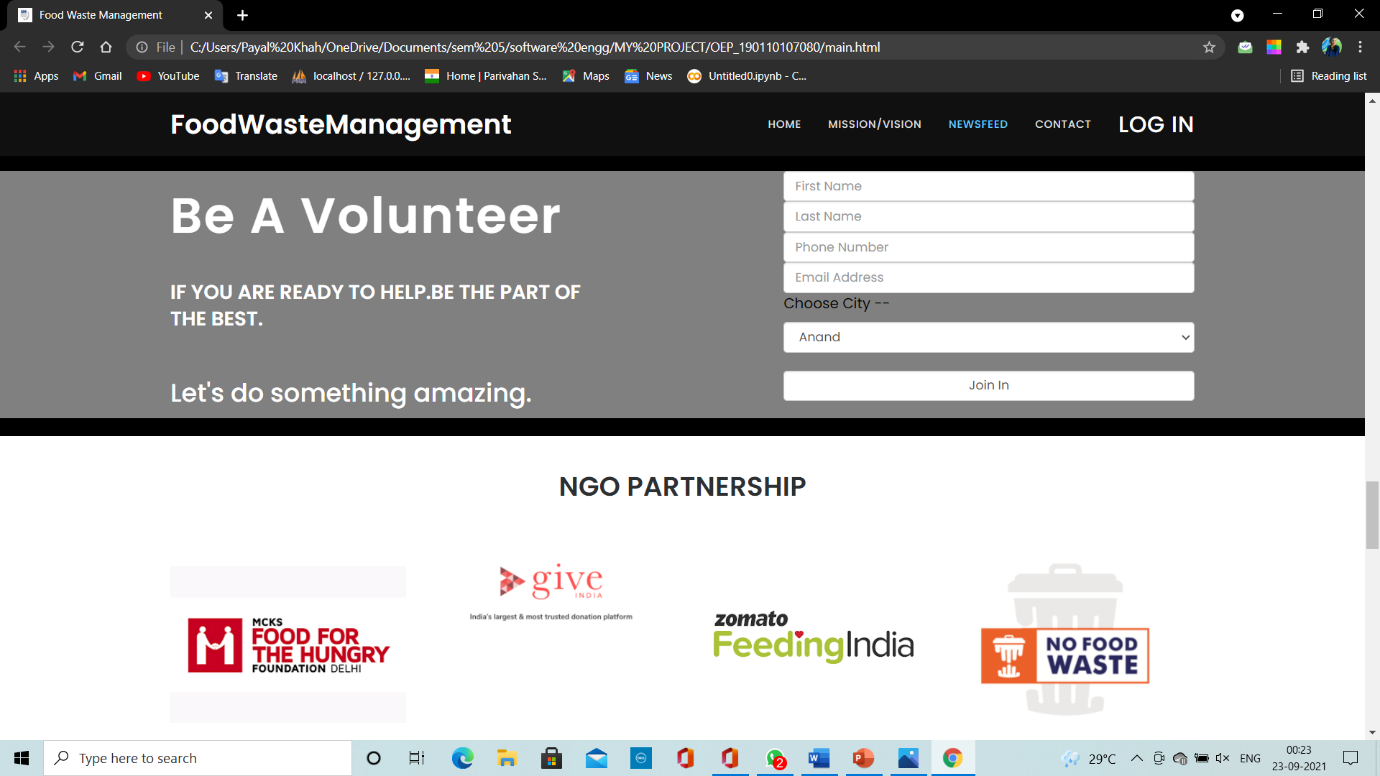


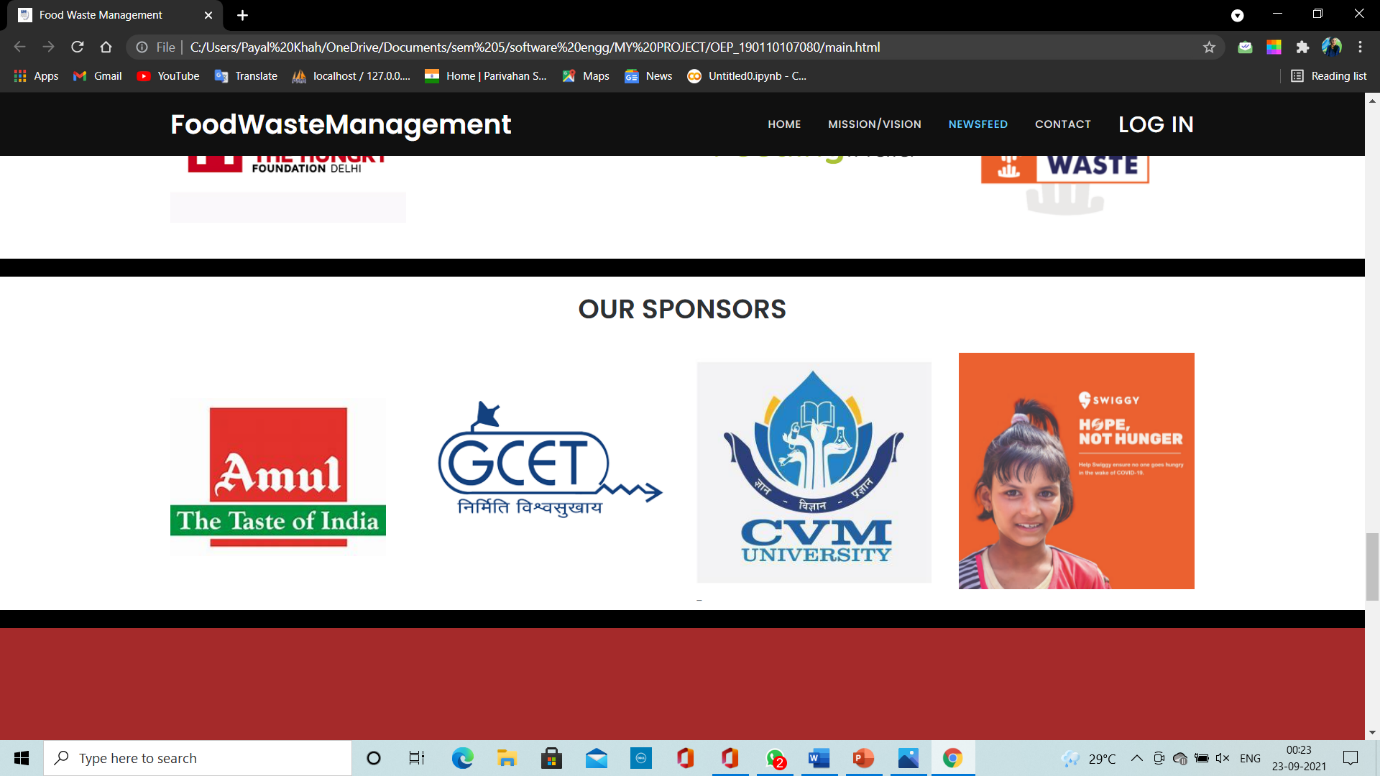


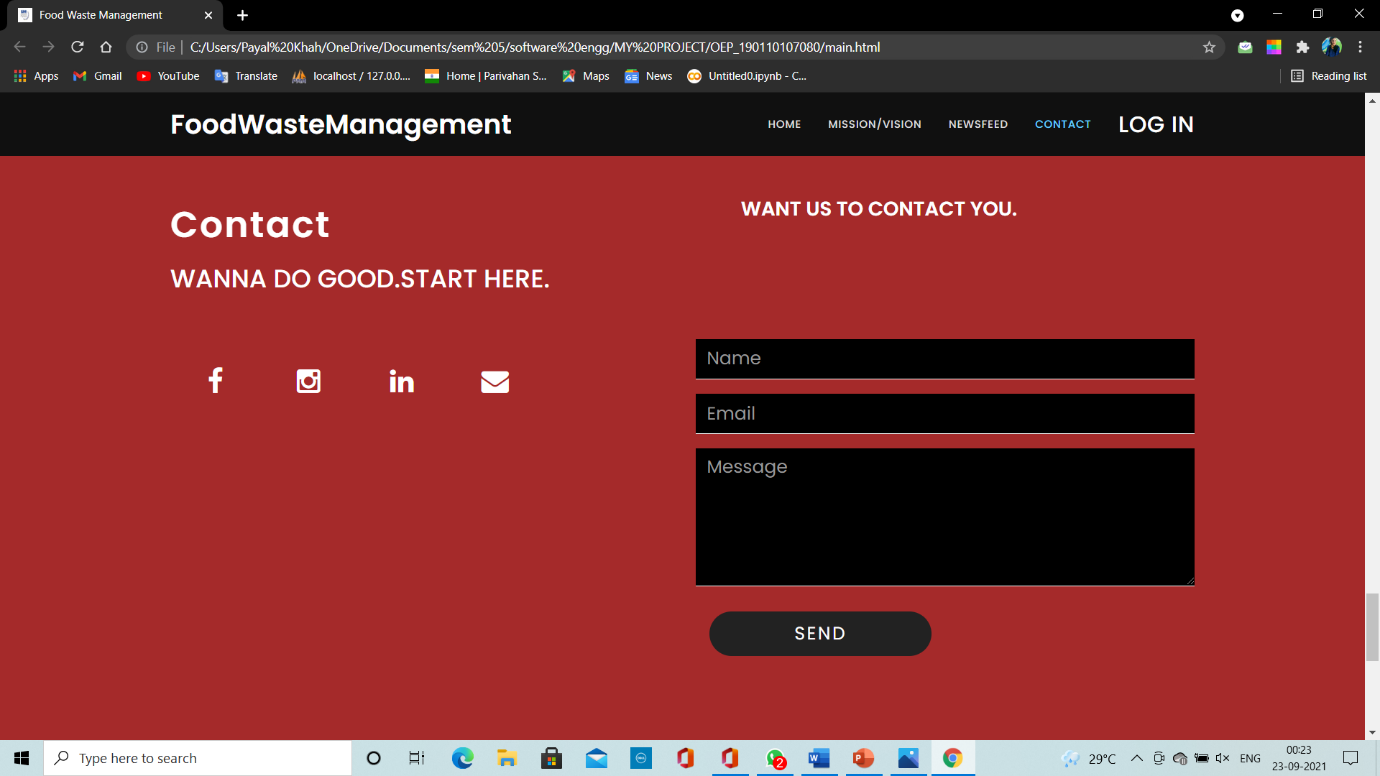


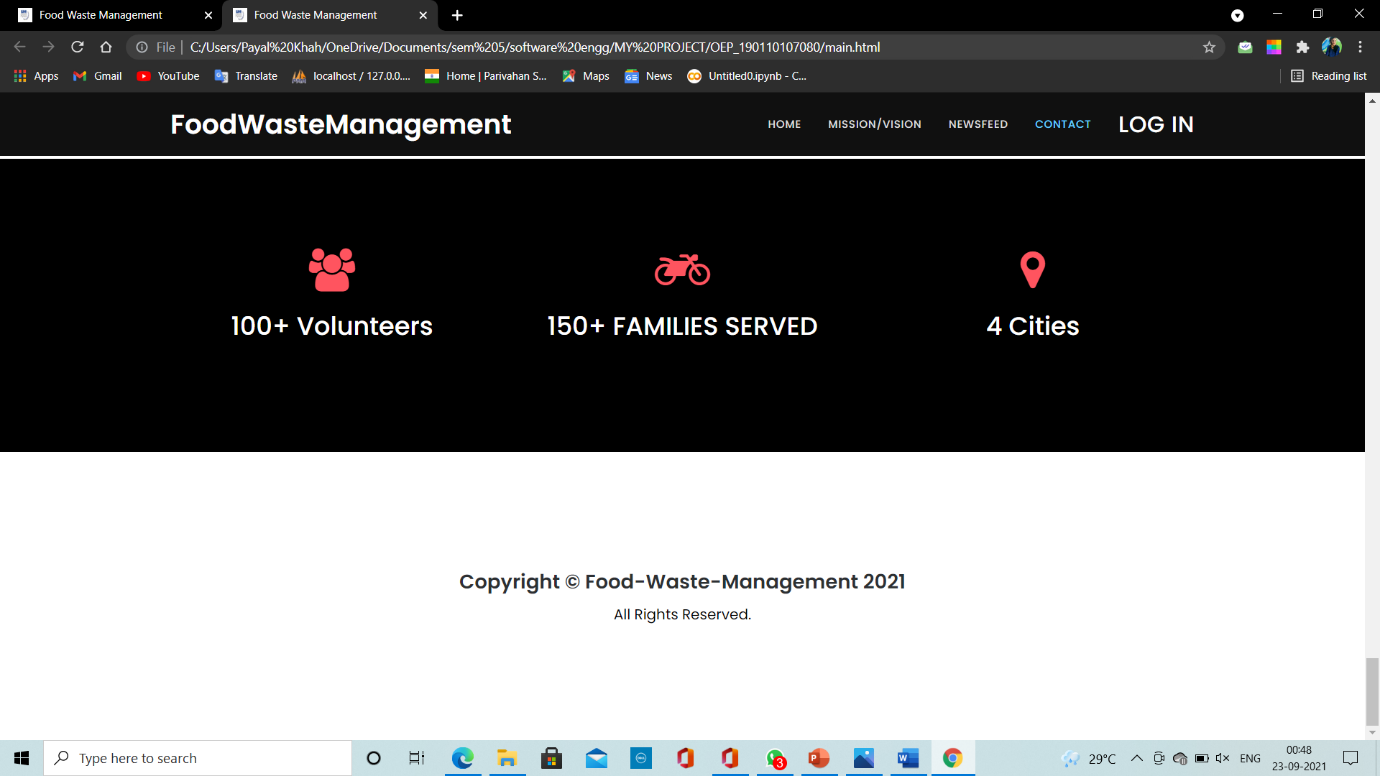


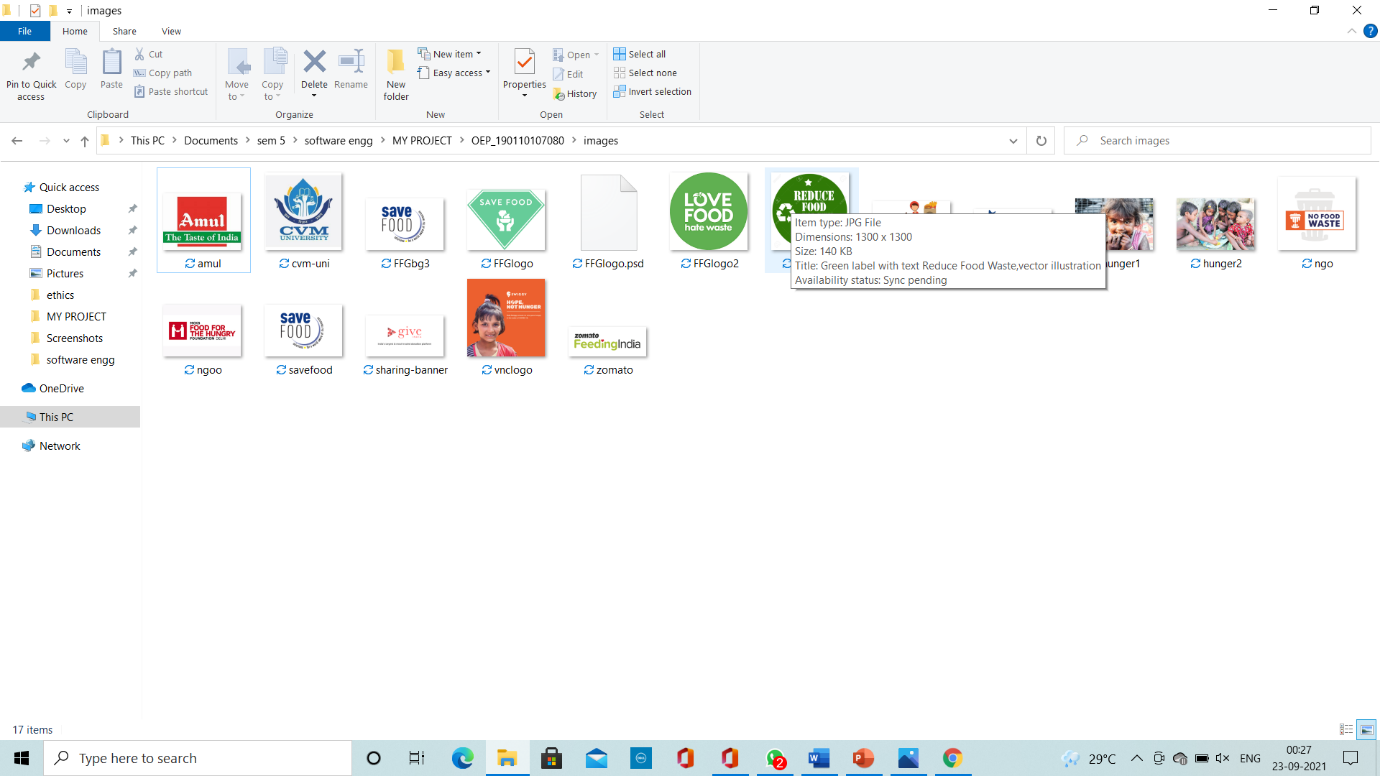




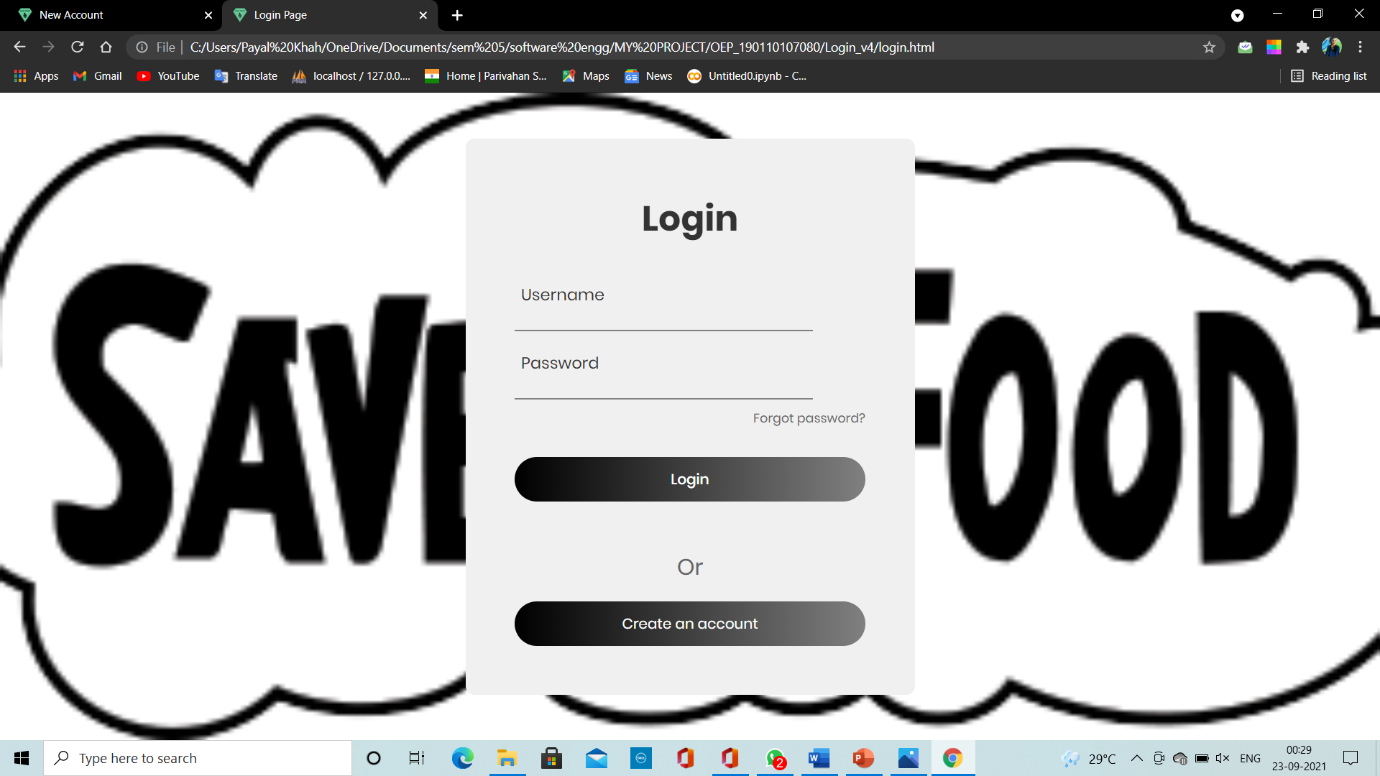


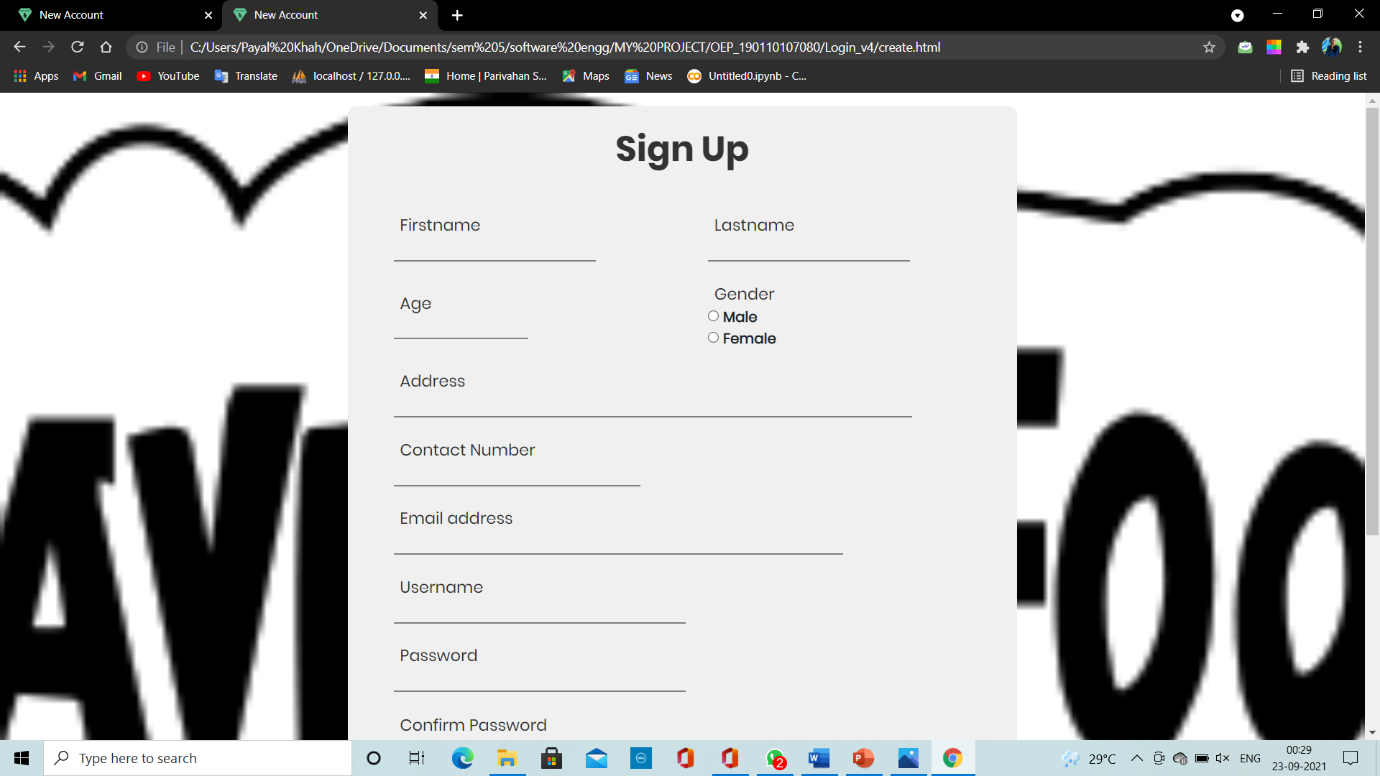


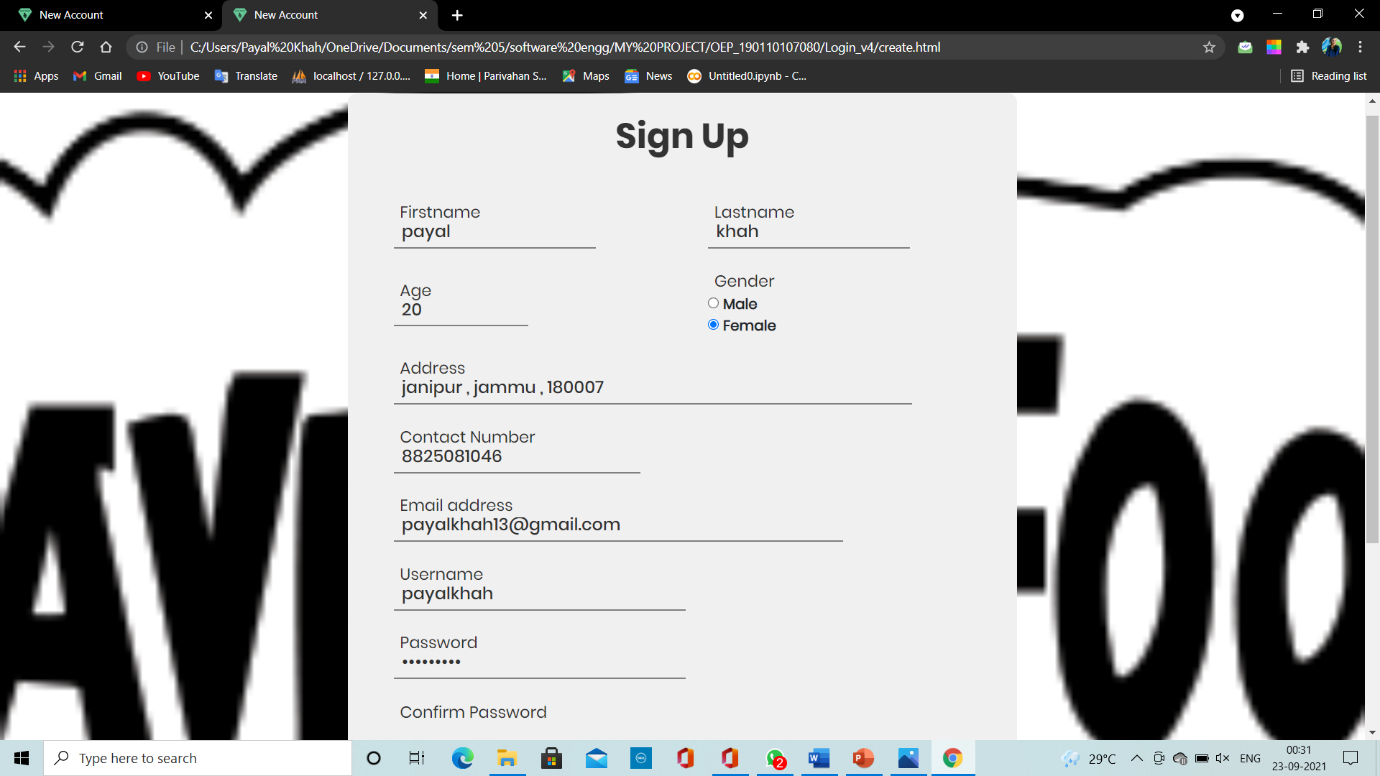


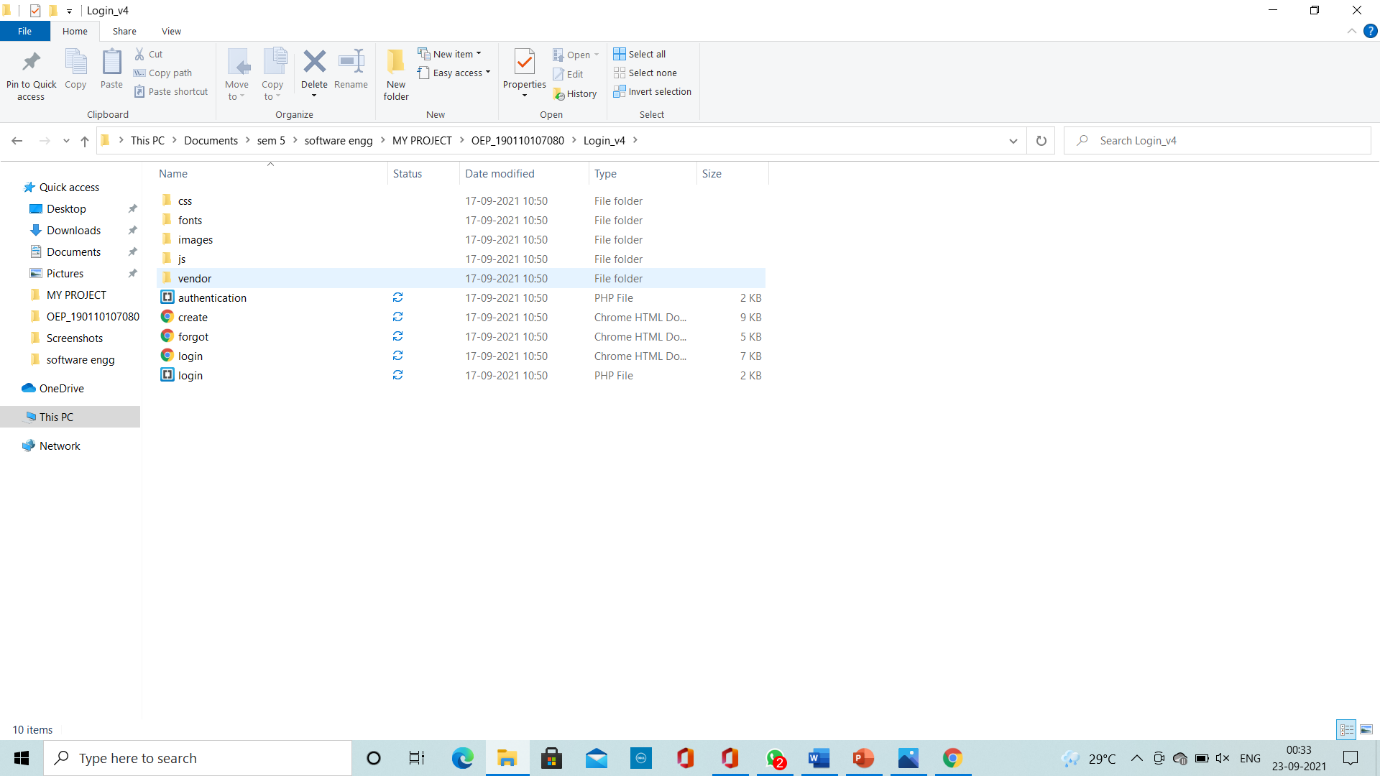


***LOGIN AND SIGNUP***









**THANK YOU….!!!!**