**SRS (Software Requirements Specifications)**

**“Developing a website for a Restaurant”**

**Idea**

Due to increase in the food cafes and restaurants, but still not having a proper staff, where the owner himself does many tasks from taking to order to billing, a website that can take order, sum up the bill, take feedbacks and convey it to the chef / owner (tentative - scalable), put on a payment gateway (tentative - expandable) so to make it convenient to the user to track his / her food that what has been ordered. Customers would be having a QR code on their respective table. They would just need to scan the QR and it would take them to the web-app / website. In that website, without even logging in, they would have the entire menu providing the facility to order the item, its quantity, and summing up the price at the end. After placing the order, it would be sent to the owner's desk, with the table number, so that it would be easier for him to serve and finally the customer would have the option to pay the bill by cash or card / upi (payment gateway - tentative).

**Technology to be used** :-

* HTML
* CSS
* JavaScript
* UI/UX Designing Tool
* APIs
* Backend/Server side technology - Java/Python/PHP (tentative)
* Database - MySQL/PopSQL/MongoDB (tentative)
* UPI plugin (tentative)

**Functional And Nonfunctional Requirements in the software.**

Functional Requirements are the requirements that make the software workable. Missing these requirements in the software would lead to non-working of the software. These are mandatory to be present for the working of the software. Functional Requirements of the problem statement in the previous experiment are:

**Functional Requirements**

* Displaying the Menu (UI)
* Displaying the order for further edits/addons
* Summing up of the order and generating a total price
* Feedback writing, storing and displaying.
* Sending the order to the admin/manager/server’s desk
* Generating a bill
* Payment gateway (optional/tentative)

Nonfunctional Requirements are complimentary requirements to the functional requirements. These requirements make the software more productive. These are not mandatory to be present for the working of the software, but if present can make it more productive to perform any task. Nonfunctional Requirements of the problem statement in the previous experiment are:

**Non - Functional Requirements**

* Scope - For the small scale restaurants and cafes having less people for management can use it as a web-app to lesser their roles.
* Scalability - In case of more restaurants, it can be turned into a community, in which every restaurant can have their profile to put on offers and ratings. Also it can be turned into cloud kitchens.
* Expandable - the tentative tasks performed which require the server side/admin side/backend can be put on when there is a lot of rush or a scarcity of manpower.
* Security - The system would not be collecting any data from the customers or users so that they can easily rely on the website without any worry of any data loss or hacking of the website.
* Usability - Easy and clear UI/UX to use for both customers and owner
* Reliability - The website would be available at every instance so customers can use it any time.