SWITO

(A Canteen Management System)

Submitted in partial fulfillment of the requirements

For the award of the degree of

Bachelor of Computer Applications (BCA)

To

Guru Gobind Singh Indraprastha University, Delhi

Guide:

Ms. Ankita Gupta

Assistant Professor

Submitted By:

Ark Bhargava (01513702020)

Khushi Panchal (03513702020)

Institute of Information Technology and Management

New Delhi – 110058

Batch (2020 – 2023)

Certificate

We, Ark Bhargava and Khushi Panchal certify that The Minor Project Report (BCA – 356) entitled "SWITO (A Canteen Management Application)" is done by us and it is an authentic work carried out by us at Institute of I formation Technology and Management. The matter embodied in this project work has not been submitted earlier for the award of any degree of diploma to the best of my knowledge and belief.

1. Ark Bhargava
2. Khushi Panchal

Date:

Certified that the Minor Project Report (BCA-357) entitled "Canteen Management System" done by the above students is completed under my guidance.

Signature of Guide:

Date:

Ms. Ankita Gupta

(Assistant Professor)

Countersigned

Director

TABLE OF CONTENTS

LIST OF FIGURES

Chapter -1 Introduction

1.1 Brief Description of system

The traditional canteen management system was suffering from a series of drawbacks. Sincewhole of the system was to be maintained with hands the process of keeping, maintaining andretrieving the information was very tedious and lengthy. The records were never used to be in a systematic order there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to gothrough the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering recordsand retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records. The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business .For this reason we have provided features Present system is partially automated (computerized), actually existing system is quite laborious as one has to enter same information at three different places.

Following points should be well considered:

- Documents and reports that must be provided by the new system: there can also be few reports, which can help management in decision-making and cost controlling, but since thesereports do not get required attention, such kind of reports and information were also identifiedand given required attention.
- Details of the information needed for each document and report.
- The required frequency and distribution for each document.
- Probable sources of information for each document and report.
- With the implementation of computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So, the proposed system helps in saving the time indifferent operations and making information flow easy giving valuable reports.

1.2 About the Proposed System

Swito, a Canteen Management App and Website is for small owners and managers of cafeteria, it digitalizes the present system of canteen management and takes it to a whole new level. It offers owners of canteens and small cafeteria to take their business online, this project contains both website and mobile application, in our website the canteen owners can registertheir canteen and soon after successful registration they will get access to admin panel from where they will be able to upload all the required details which will be reviewed by our teamand then it will be made available for the customers, owners can only register or login to admin panel through ourwebsite.

The purpose of Swito is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Swito, as described above, can lead to error free, secure, reliable and fast management system.It can assist the user to concentrate on their other activities rather to concentrate on the recordkeeping. Thus it will help organization in better utilization of resources. The organization canmaintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

The main objective of the Project on Canteen Management System is to manage the details of Canteen, Snacks Type, Canteen Staff. It manages all the information about Bill Payment. The purpose of the project is to build an application program to reduce the manual work for managing the Bill Payment, consumer Meal. It tracks all the details about the customer Meal, Meal Type, Canteen menu.

Functionalities provided by Swito are as follows

- Provides the searching facilities based on various factors. Such as Canteen name.
- Swito also manage the Bill Payment details online for Snacks Type details.
- It tracks all the information of Customer, Bill Payment, Snacks Type.
- Shows the information and description of the Canteen, Customer Snacks.
- To increase efficiency of managing the Company Canteen, Employee
- It deals with monitoring the information and transactions of Meal Type.
- Manage the information of Company Canteen
- Editing, adding and updating of Records is improved which results in proper resourcemanagement of Canteen data.
- Manage the information of Canteen Menu.

Scope of the Project

It may help collecting perfect management in details. In a very short time, the collection willbe obvious, simple and sensible. It will help a person to know the management of passed yearperfectly and vividly. It also helps in current all works relative to Canteen Management System. It will be also reduced the cost of collecting the management & collection procedurewill go on smoothly. Our project aims at Business process automation, i.e. we have tried to computerize various processes of Swito.

Input Data and Validations of project Swito

- All the fields are validated and does not take invalid values
- Each form for Swito, Customer, Bill Payment cannot accept blank value fields
- Avoiding errors in data
- Controlling amount of input
- Integration of all the modules/forms in the system.
- Preparation of the test cases.

- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.
- Testing the module with all the possible test data.

1.3 Methodology of System Design

SDLC Model used

The Software Development Lifecycle is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software which meets customer expectations. The software development should be complete in the pre-defined time frame and cost. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC lifecycle has its own process and deliverables that feed into the next phase. Since our project deals with creation of a website that has basic modules first and then furthermodule with changes are made, so incremental model is being used here.

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release untilall designed functionality has been implemented. Incremental model is also known asSuccessive version model.

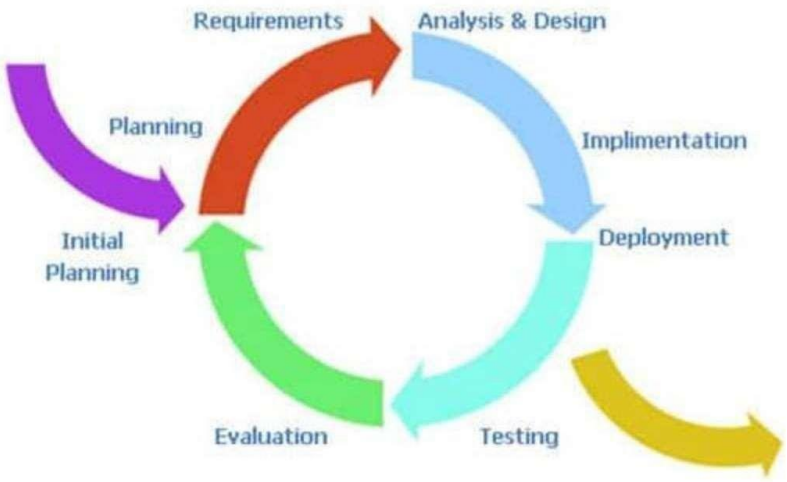


Fig 1.1 Incremental Model

1. **Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. **Design & Development**: In this phase of the Incremental model of SDLC, the design ofthe system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. **Testing:** In the incremental model, the testing phase checks the performance of each existingfunction as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.
4. **Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

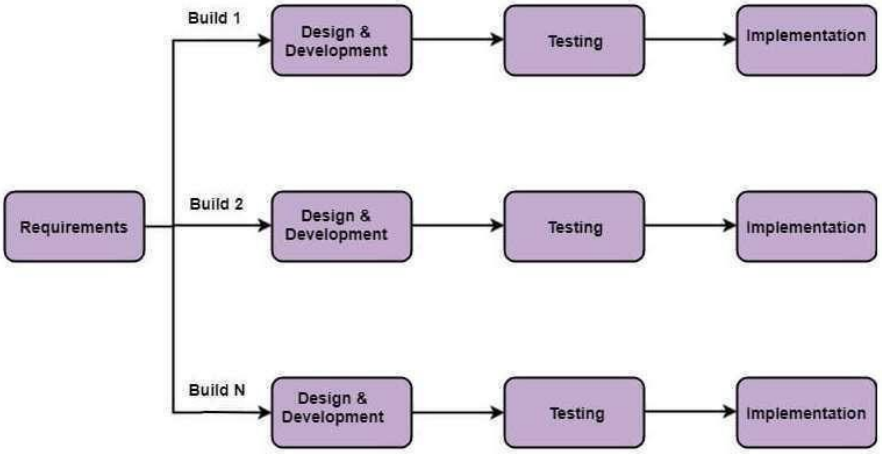


Fig 1.2 Incremental Model Phase

Characteristics of incremental model

- System development is broken down into many mini development projects
- Partial systems are successively built to produce a final total system
- Highest priority requirement is tackled first
- Once the requirement is developed, requirement for that increment are frozen

Advantages of Incremental Model

- Generates working software quickly and early during the software life cycle.
- More flexible – less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.
- Each iteration is an easily managed milestone.

As our Project contains both mobile application so we have designed the user interfaces first and then connected it to a single backend and in this process we have adopted Incremental Model of SDLC.

1.4 Methodology used for Data Collection

Data collection is a process of collecting information fromall the relevant sources to find answersto the research problem, test the hypothesis and evaluate the outcomes. Data collection is a fundamental aspect and as a result, there are different methods of collecting data which when used on one particular set will result in different kinds of data. Collection of data refers to a purpose gathering of information and relevant to the subject-matter of the study from the units under investigation. The method of collection of data mainly depends upon the nature, purpose and the scope of inquiry on one hand and availability of resources, and the time to the other.

The statistical Data may be classified into primary and secondary depending upon the natureand mode of collection.

1.4.1 Primary Data Collection Methods

Primary data is always collected from our research instead of books or any other written material,which come from second-hand sources. It is original in nature and is specific to a research problem under study. The various methods of primary data collection are :

- Interview
- Questionnaire
- Schedule Method
- Projective Technique
- Focus Group Interview

1.4.2 Secondary Data Collection Methods

The secondary data are readily available from the other sources and as such, there are no specificcollection methods. The researcher can obtain data from the sources both internal and external tothe organization. The internal sources of secondary data are:

- Sales Report
- Customer details, like name, age, contact details, etc.
- Company information
- Reports and feedback from a dealer, retailer, and distributor
- Management information system

1.5 System Requirements tools

Hardware Requirements

RAM- 2 GB & Above

Hard Disk- 32 GB min

Processor: Minimum 1 GHz Recommended 2GHz or moreEthernet connection (LAN) OR a wireless adapter (Wi-Fi)

Operating System- Windows OS, MAC OS, Linux OS

Frontend tool for web Application

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's componentsclearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

AngularJS is what HTML would have been, had it been designed for applications. HTML is agreat declarative language for static documents.

Backend Tool for Web Application

Mongo DB is an open-source database management system (DBMS) that uses a document-oriented database model which supports various forms of data.

It is one of numerous non-relational database technologies which arose in the mid-2000s under the No-SQL banner for use in big data applications and other processing jobs involvingdata that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the Mongo DB architecture is made up of collections and documents.

1.6 Project Planning

Project planning is a procedural step in project management, where required documentaon is created to ensure successful project completion. Documentation includes all actions required to define, prepare, integrate and coordinate additional plans. The project plan clearly defines how the project is executed, monitored, controlled and closed.

The basic processes of project planning are:

- Scope planning – specifying the in-scope requirements for the project to facilitate creating thework breakdown structure
- Preparation of the work breakdown structure – spelling out the breakdown of the project intotasks and sub-tasks
- Project schedule development – listing the entire schedule of the activities and detailing theirsequence of implementation
- Resource planning – indicating who will do what work, at which time, and if any special skillsare needed to accomplish the project tasks
- Budget planning – specifying the budgeted cost to be incurred at the completion of the project
- Procurement planning – focusing on vendors outside your company and subcontracting
- Risk management – planning for possible risks and considering optional contingency plans andmitigation strategies
- Quality planning – assessing quality criteria to be used for the project
- Communication planning – designing the communication strategy with all project stakeholder

System design of canteen management system

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients's requirements into a logically workingsystem. Normally, design is performed in the following in the following two steps:

1. Primary Design Phase: In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification

phase. Different blocks are created for different functions emphasis is put on minimizing the information flow betweenblocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design Phase: In the secondary phase the detailed design of every block isperformed.

Project Scheduling

Gantt Chart is also known as Bar chart is used exclusively for scheduling purpose. It is a projectcontrolling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is abar chart with each bar representing activity. The bars are drawn against a time line. The length of time planned for the activity.

An elementary Gantt chart or Timeline chart for the development plan is given below. The plan explains the tasks versus the time (in weeks) they will take to complete.

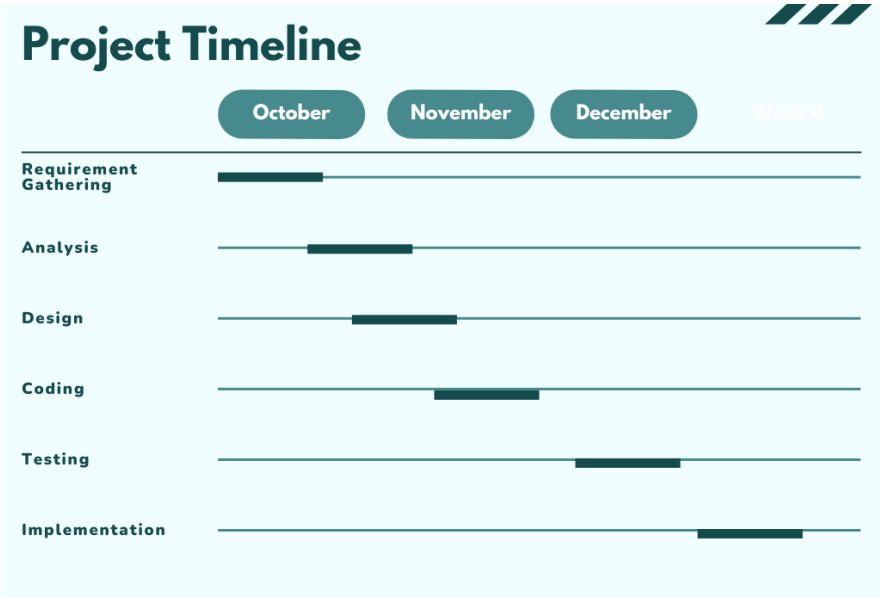


Fig 1.3 Gantt Chart

Chapter – 2 System Requirement Specifications

System Requirement Submission Report (As per IEEE Format): Prepare a SRS document as per IEEEstandard notation for implementing the user requirement of the proposed system.

A software requirements specification (SRS) is a description of a software system to be developed. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document lists sufficient and necessary requirements for the project development.to derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

2.1 Purpose Scope

The main purpose of Software Requirement Specifications Document is to describe in a precise manner all the capabilities that will be provided by the Software Application “SWITO (Canteen Management)”. It also states the various constraints which the system will be abide to. This documentfurther leads to clear vision of the software requirements, specifications and capabilities. These are tobe exposed to the development, testing team and end users of the software.

The objective of our project is to provide a user interactive website and mobile applications for our users and adding new techniques day-by-day also we have provided a feedback form to know how much our customers and users are satisfied and in which areas we need to work.

2.1.1 Definitions, acronyms & abbreviations

Swito application is an application where customer can book their snacks online instead of waiting inline just by click of a button.

Swito uses Mongo DB database which is a non-SQL database that provides built in data management,data access and data manipulation operations.

Abbreviations used in this project:-

MongoDB- This is a SQL database. In this data are stored in form of collections and in everycollection, we have documents.

NodeJS- This is known as Node JavaScript. This is used for server-side scripting.

2.1.2 References

1. ReactNative.org
2. Github.org
3. YouTube tutorials
4. Stackoverflow.com
5. MongoDB.com

2.1.3 Overview

This project aims to eliminate the disadvantages and problems we face in our daily lives to give ordersin canteens and cafeteria, such as waiting for a long time, paying money in cash and deciding menu, inprofessional work place the snacks breaks are for very short duration, so, with the help of this project it will be very efficient for both customers and owners to manage theirtask.

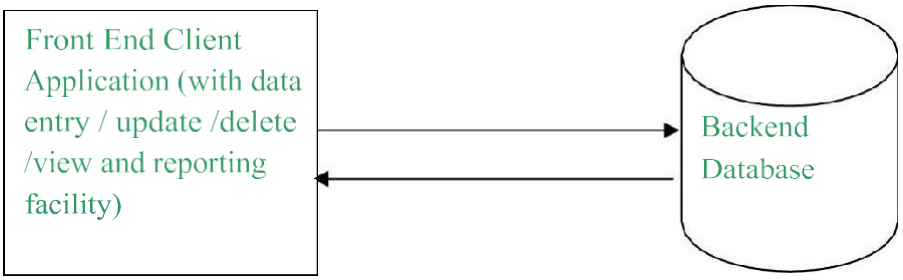
The rest of this SRS document describes the various system requirements, interfaces, features andfunctionality in detail.

2.2 System Requirement specifications

2.2.1 Product Perspective

Our project has the following feature-

- To buy any snacks online, the customer just has to use our mobile application.
- The customer does not require to register themselves for ordering snacks, they just have togive their details like name, Paytm number while ordering the snacks.
- It is very busy to book snacks online by the help of Swito. The customer just has to select the item for the given list and the quantity, bill will generate automatically.
- We have used two payment option in this project first is Paytm and second one is COD.
- Our project aim is to digitalize the old canteen management system.
- Our project deals with online database that has its own features to access and maintaindata. So multiple customers can order snacks online.



2.2.2 Overall Description of the proposed system

SWITO, a Canteen Management App and Website is for small owners and managers of cafeteria,it digitalizes the present system of canteen management and takes it to a whole new level . It offers owners of canteens and small cafeteria to take their business online , this project containsboth website and mobile application ,in our website the canteen owners can register their canteen and soon after successful registration they will get access to admin panel from wherethey will be able to upload all the required details which will be reviewed by our team and then it will be made available for the customers ,owners can only register or login to admin panel through our website .

The system design of our website is very simple and less complex, so that user can understand it easily. Our project only requires a desktop or laptop and an internet connection.

The various functionalities include :

- It includes all the basic modules first that is – Login/ Registration.
- The user either has to login or register before purchasing any food from the website.
- The user first gets to see the home page of our website.
- There can be single administrator on this website.
- Once the details are entered, the data is saved at the backend database. It will be usedin future for the customer to login whenever he wants to.
- The customer once logged in, then can purchase food he wants to and can use his loginid and password in future again to login whenever he wants to.

2.2.3 Hardware and Software Specifications

Most hardware devices also include a user interface, though it is typically not as complex as asoftware interface. A common example of a hardware device with a user interface is a remotecontrol. A typical TV remote has a numeric keypad, volume and channel buttons, mute and power buttons, an input selector, and other buttons that perform various functions. This set ofbuttons and the way they are laid out on the controller makes up the user interface. Other devices, such as digital cameras, audio mixing consoles, andstereo systems also have a user interface.

While user interfaces can be designed for either hardware of software, most are a combination ofboth. For example, to control a software program, you typically need to use a keyboard and mouse, which each have their own user interface. Likewise, to control a digital camera, you mayneed to navigate through the on-screen menus, which is a software interface.

RAM- 3GB

ROM-32 GB S

Processor- Dual core processor or above is required.

A software interface is used to allow either two pieces of software to communicate with each other (software-software interface), or to allow software to communicate with a hardware device (software-hardware interface).

Operating system- Windows OS, MAC OS, Linux OS

Development tool- React-Native, Mongo DB, Postman, NodeJS, Angular JS

Database used- MongoDB

Chapter – 3 System Design

3.1 Block Diagram

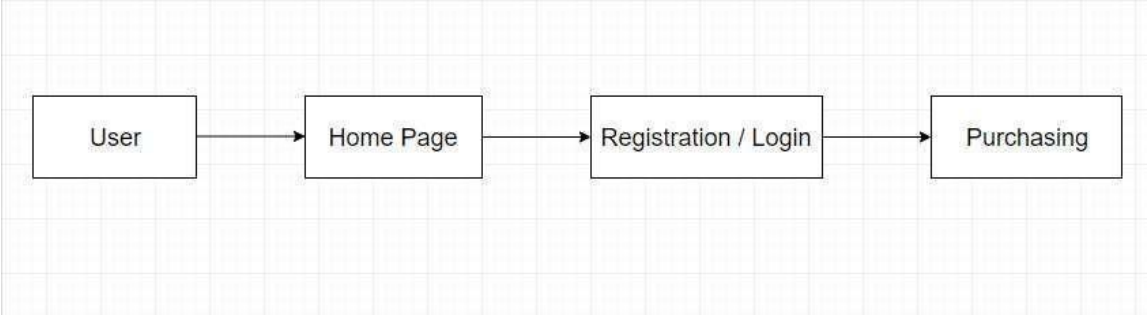
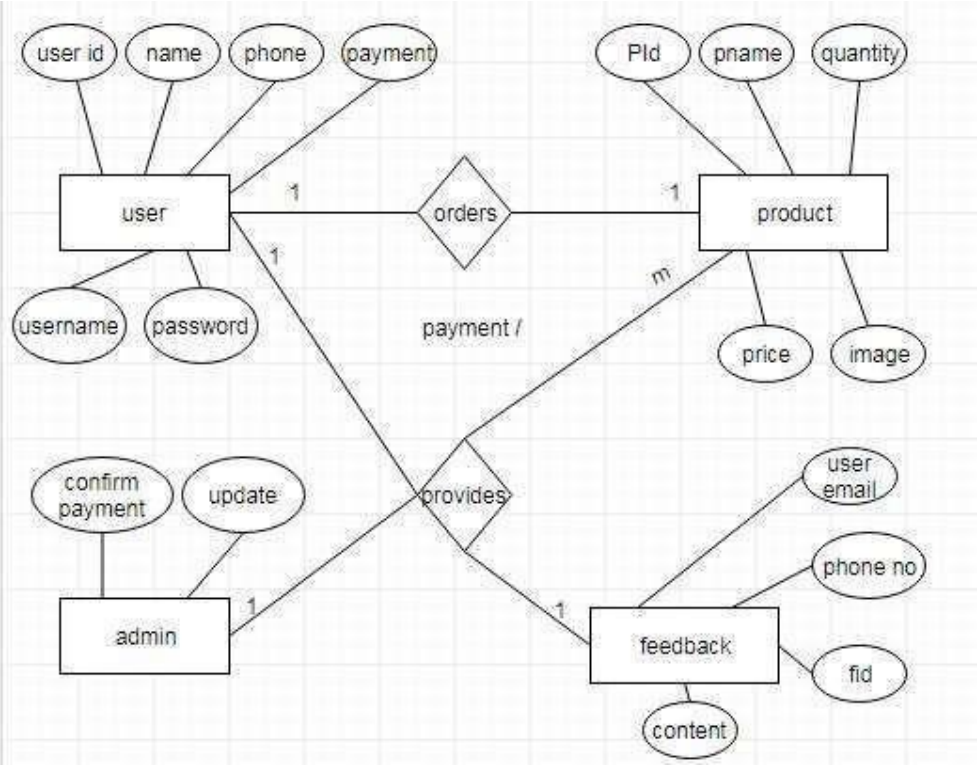


Fig 3.1 Block Diagram

3.2 Entity Relationship Diagram



3.3 Data Flow Diagram (0 level)

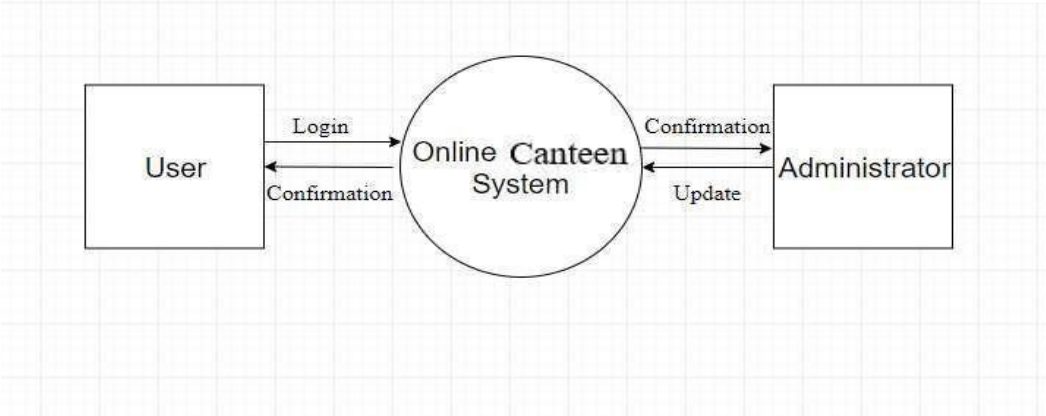


Fig 3.3 DFD (0 level)

3.4 Data Flow Diagram (1st level)

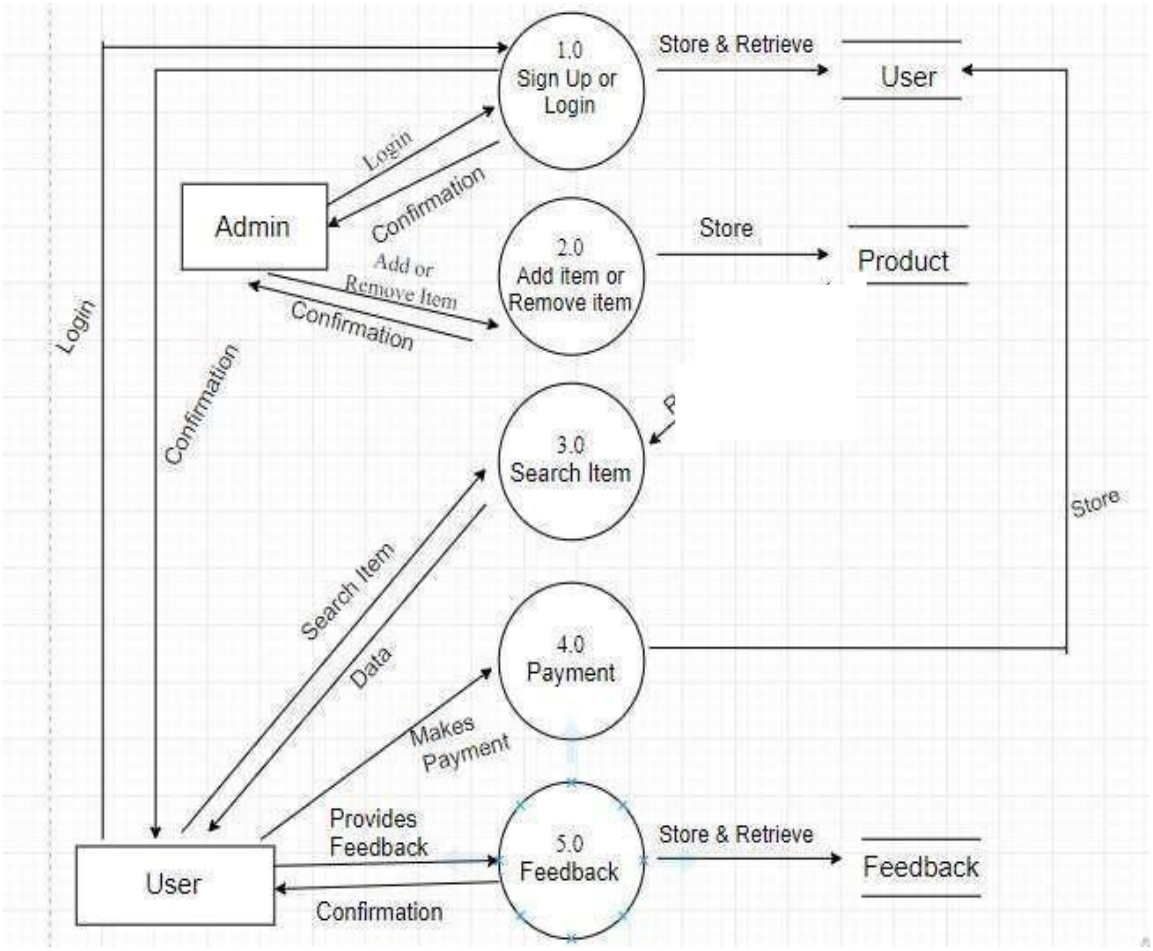


Fig 3.4 DFD (1st level)

3.5 Use Case Diagram

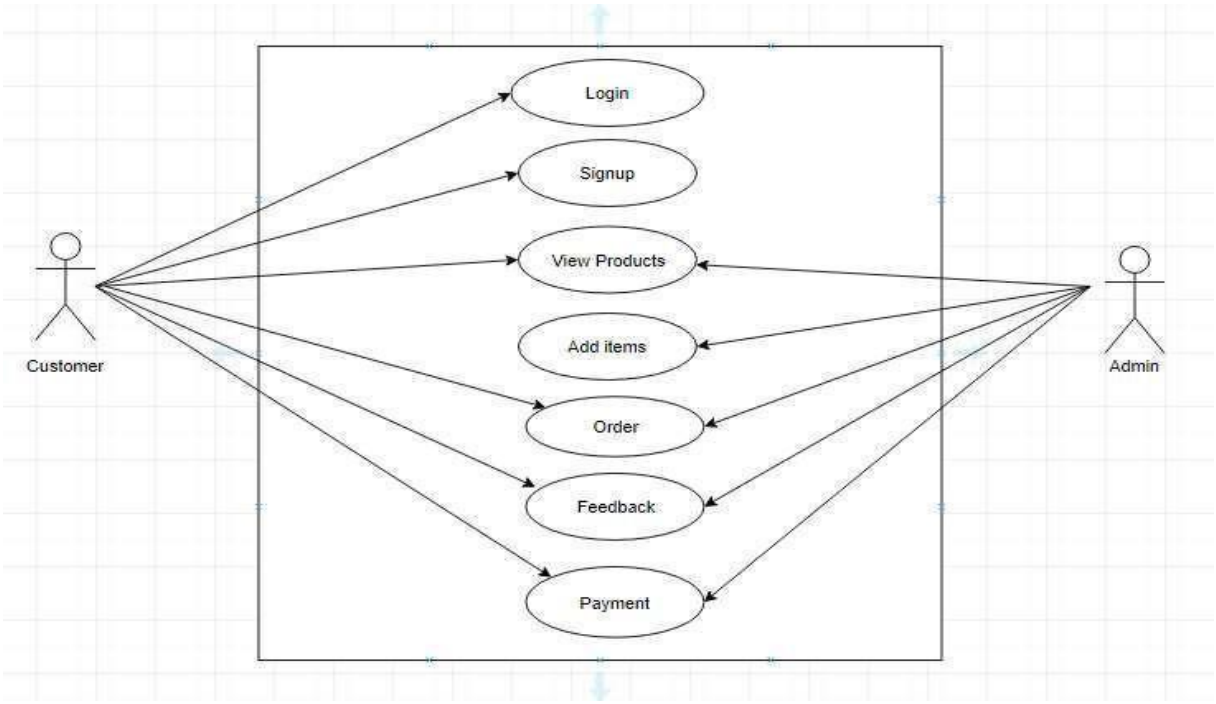


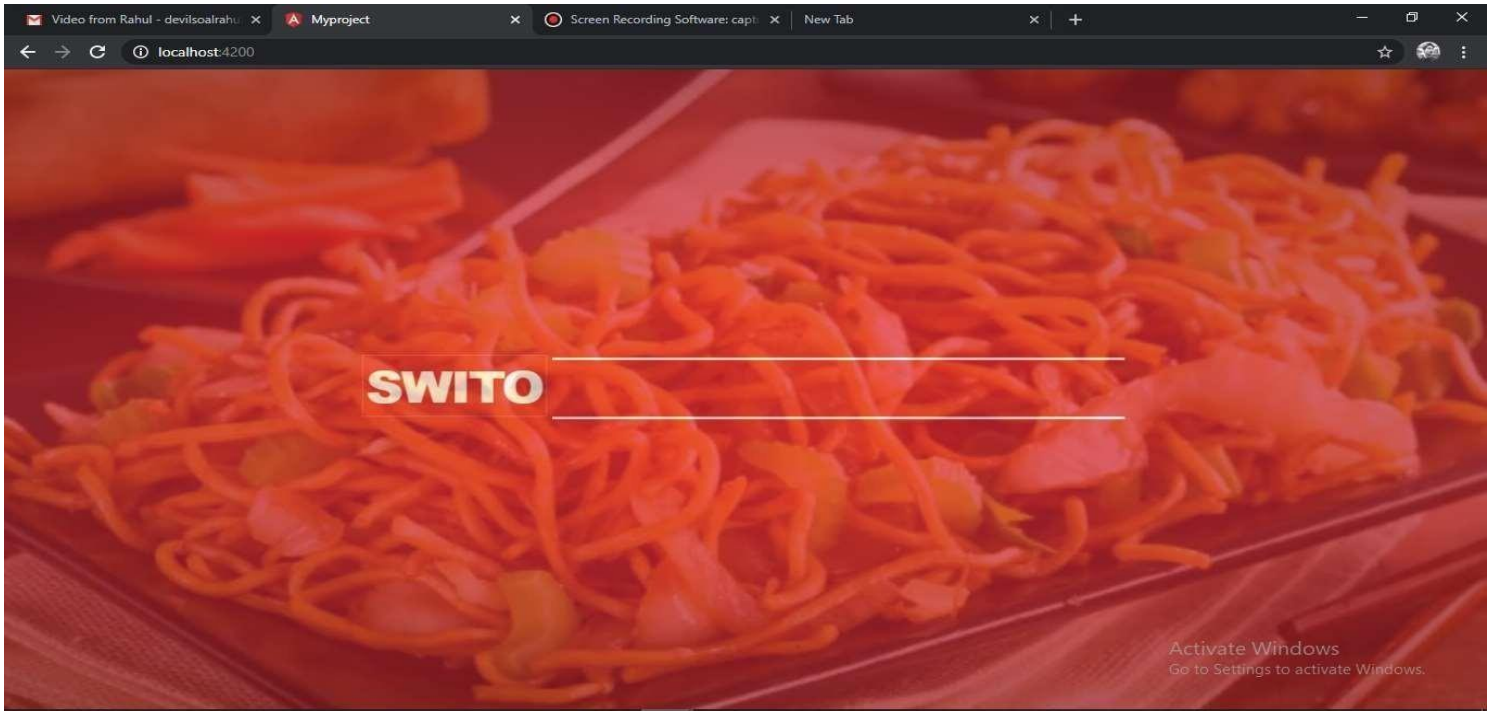
Fig 3.5 Use Case

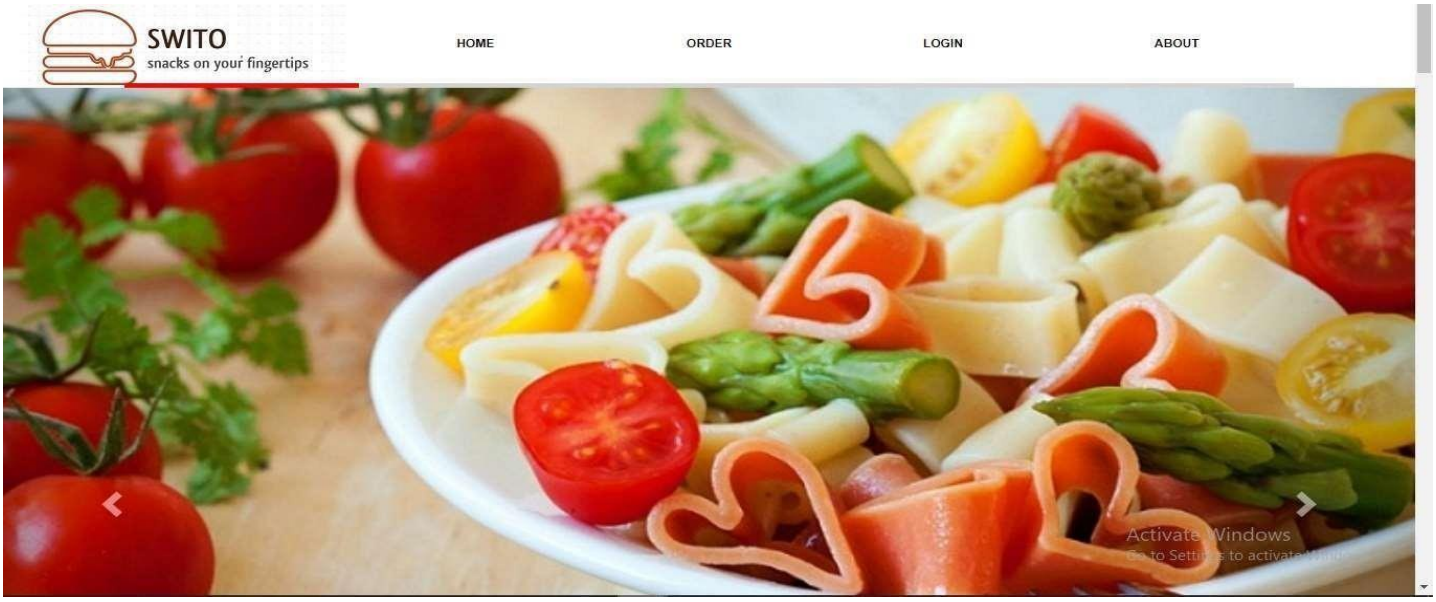
3.6 Database Design

Field name	**Field Code**	**Field Type**	**Size**	**Description**
Paytm number	Pay num	Depends on input	Depends on input	Paytm Number
Item Name	prod_name	Depends on input	Depends on input	Name of Item
Price	Price	Depends on input	Depends on input	Item price
Quantity	Qty	Depends on input	Depends on input	Quantity
Field name	**Field Code**	**Field Type**	**Size**	**Description**
Customer Address	Address	Depends on input	Depends on input	Address of customer
Customer Name	Cust name	Depends on input	Depends on input	Name of customer
Email	Email	Depends on input	Depends on input	Email
Password	Pass	Depends on input	Depends on input	Password
Phone number	Phone	Depends on input	Depends on input	Phone number
Field name	**Field Code**	**Field Type**	**Size**	**Description**
Customer Email	email	Depends on input	Depends on input	Customer email
Customer Name	name	Depends on input	Depends on input	Customer name
Contents	Cont	Depends on input	Depends on input	Feedback content

Chapter – 4 System Developments

4.1 Webpages(s) Design





4.2 Output Designs

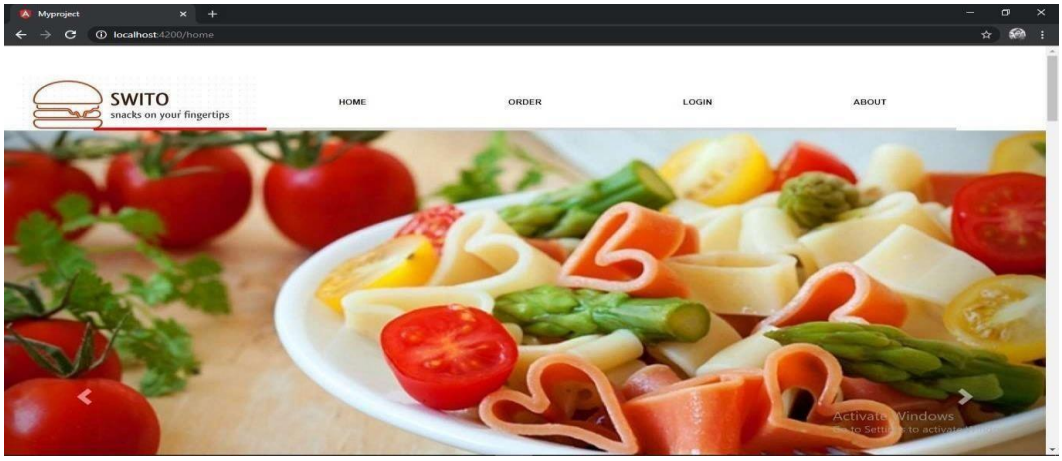


Fig 4.2.1 Home Page

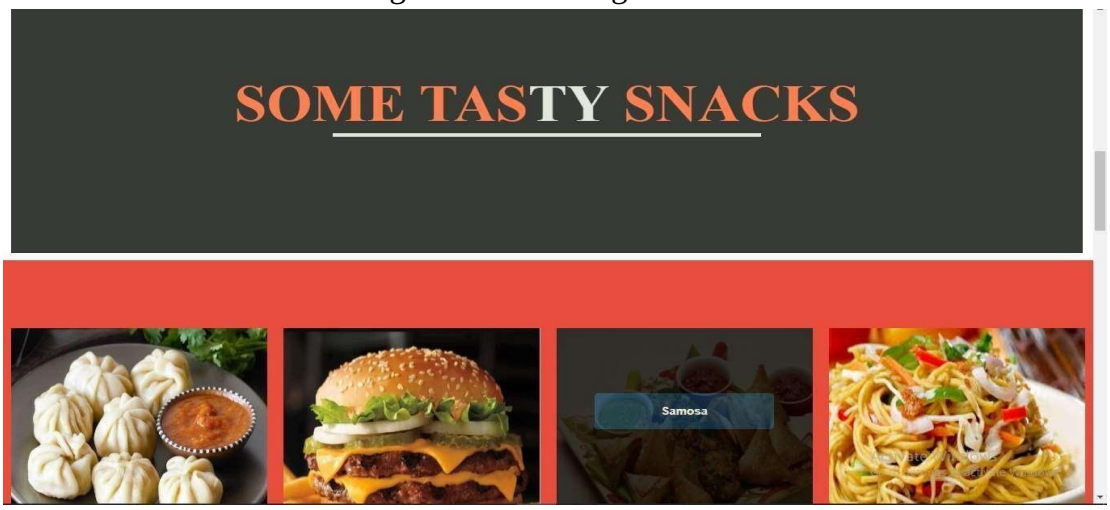


Fig 4.2.2 Items Page

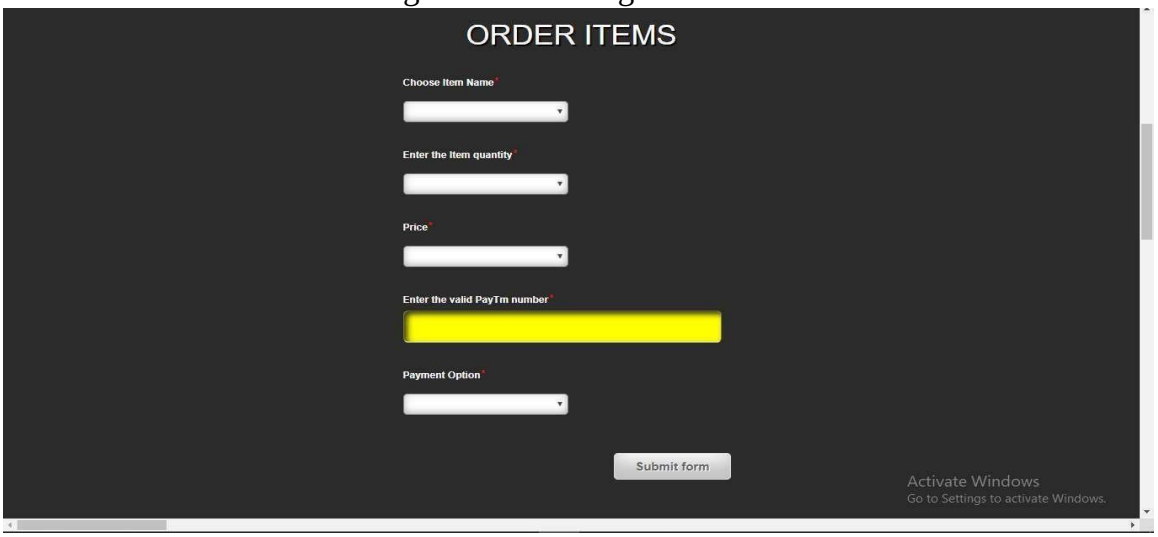


Fig 4.2.3 order Items

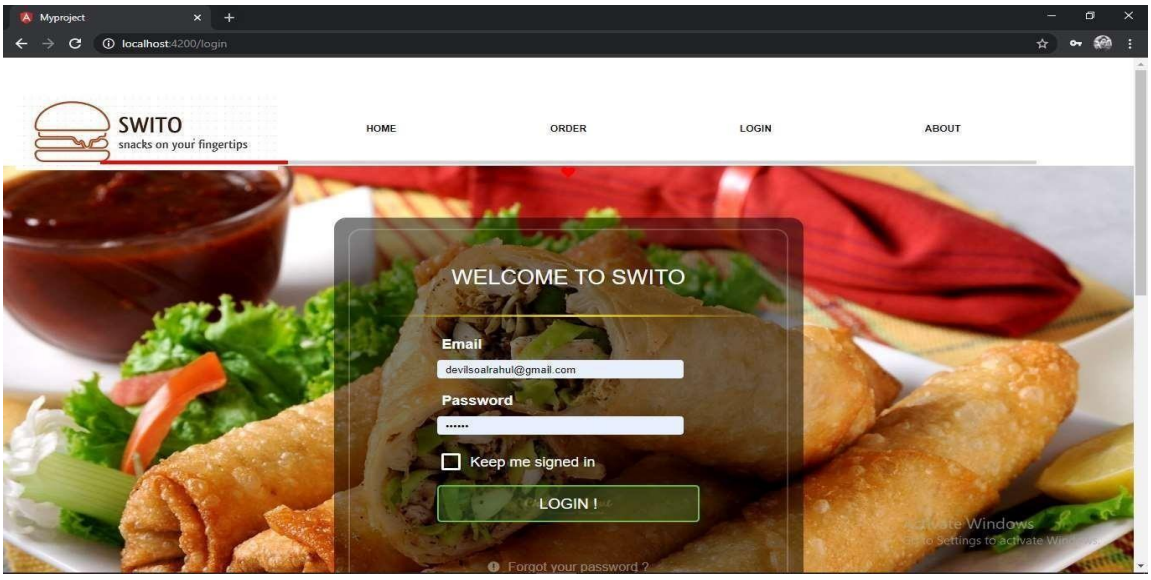


Fig 4.2.4 Login page

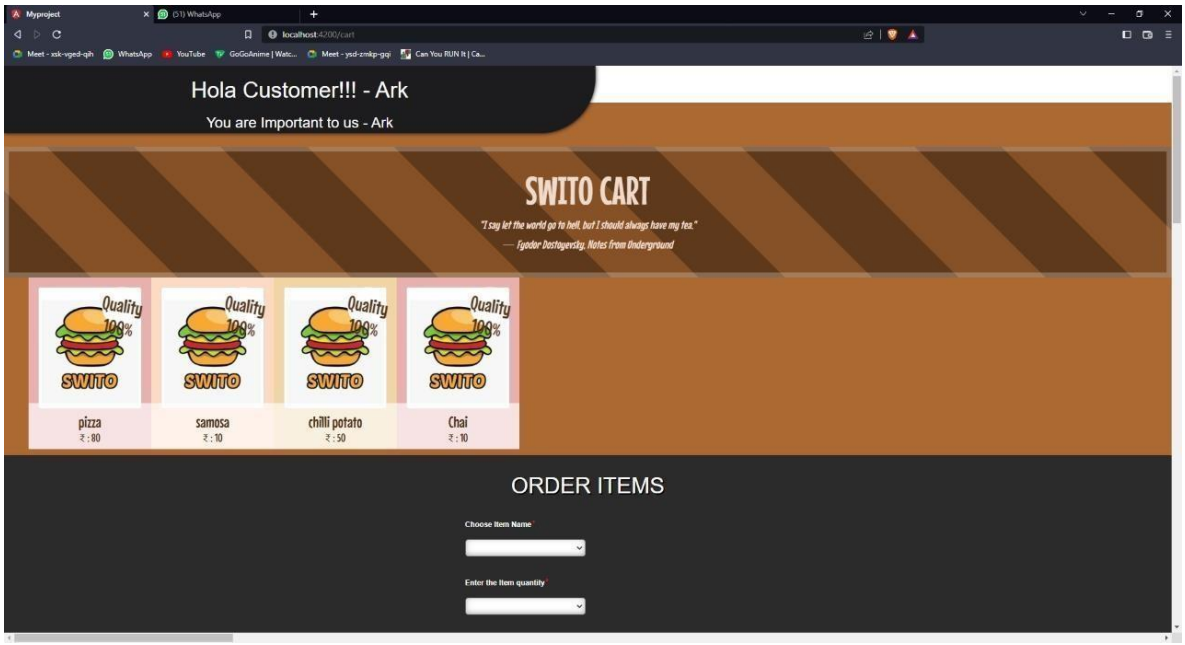


Fig 4.2.5 Order Items

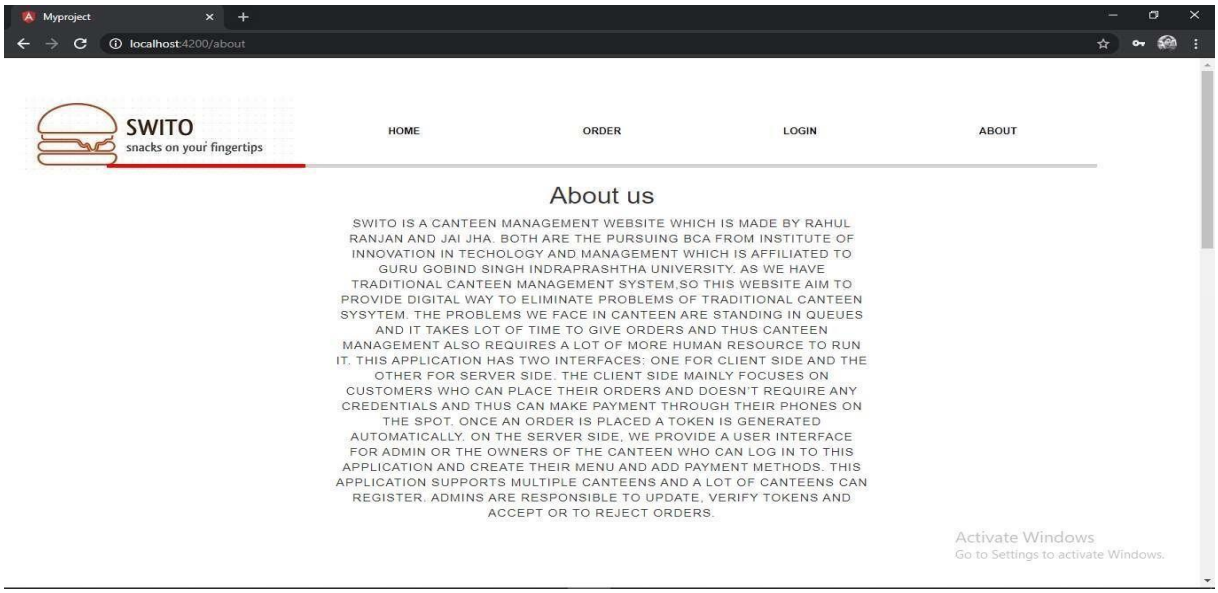


Fig 4.2.6 About us

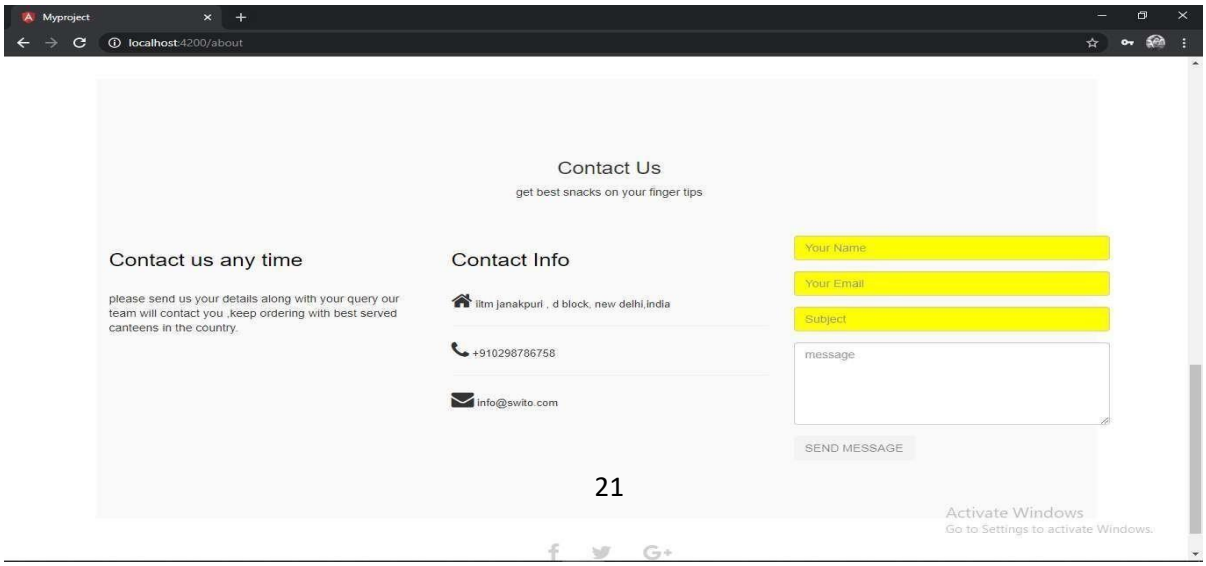
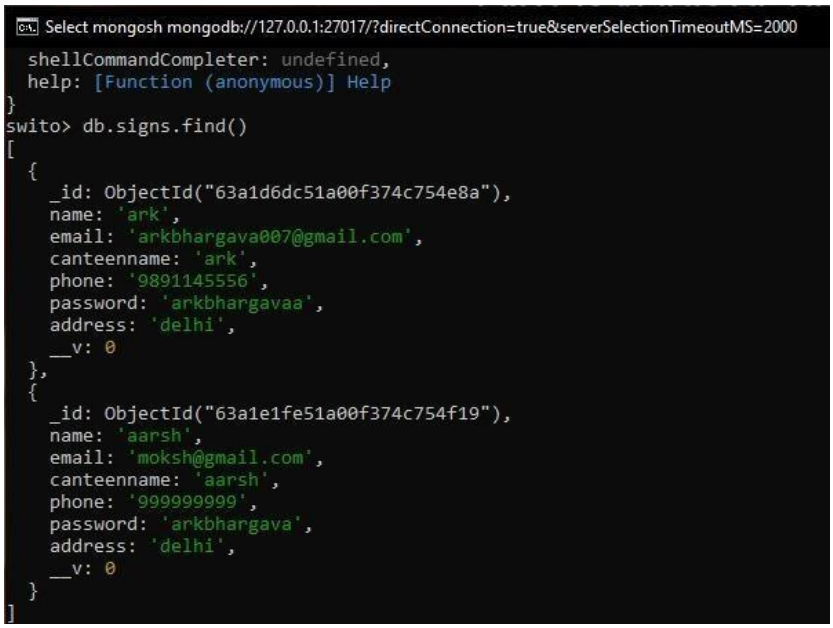


Fig 4.2.7 Contact us

4.3 Database Design



home.component.html

```
   <app-header></app-header>
   <html>
     <head>
  <script  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js"
integrity="sha384-
b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4"
crossorigin="anonymous"></script>
     </head>
     <body>
   <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>
      </ol>

      <!-- Wrapper for slides -->
      <div class="carousel-inner">
        <div class="item active">
          <img src="../../assets/chicago.jpg" alt="Chania" height=600 width=1500>
          <div class="carousel-caption">
            <h2>WELCOME TO SWITO</h2>
            <audio controls>
               <source src="../../assets/w.ogg" type="audio/ogg">
               <source src="../../assets/w.mp3" type="audio/mpeg">
              Your browser does not support the audio element.
              </audio>
          </div>
        </div>

        <div class="item">
          <img src="../../assets/ny.jpg" alt="Chicago" height=450 width=1500>

          <div class="carousel-caption">
            <h2>Delicious Snacks On Your Fingertips</h2>
            <audio controls>
               <source src="../../assets/q.ogg" type="audio/ogg">
               <source src="../../assets/q.mp3" type="audio/mpeg">
              Your browser does not support the audio element.
              </audio>
          </div>
```

```
      </div>

      <div class="item">
        <img src="../../assets/la.jpg" alt="New York"height=450 width=1500>
        <div class="carousel-caption">
          <h2>Quality Snacks At Reasonable Price</h2>
          <audio controls>
              <source src="../../assets/m.ogg" type="audio/ogg">
              <source src="../../assets/m.mp3" type="audio/mpeg">
            Your browser does not support the audio element.
            </audio>
        </div>
      </div>
    </div>

    <!-- Left and right controls -->
    <a class="left carousel-control" href="#myCarousel" data-slide="prev">
      <span class="glyphicon glyphicon-chevron-left"></span>
      <span class="sr-only">Previous</span>
    </a>
    <a class="right carousel-control" href="#myCarousel" data-slide="next">
      <span class="glyphicon glyphicon-chevron-right"></span>
      <span class="sr-only">Next</span>
    </a>
  </div>

  </body>
  </html>
     <app-heading></app-heading>
  <app-raa></app-raa>
  <app-raa2></app-raa2>
  <app-raa3></app-raa3>
<app-footer></app-footer>
```

header.component.html

```
<!-- <header id="header">
   <nav class="navbar navbar-fixed-top" role="banner">
     <div class="container">
       <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
        <a class= "navbar-brand" href="/home" >Swito</a>
       </div>
       <div class="collapse navbar-collapse navbar-right"> -->
            <!-- <nav class='navbar navbar-expand navbar-light bg-light'> -->
                 <!-- <nav id="main-navbar"> -->
                  <!-- <ul  ></ul>class="nav nav-pills"> -->
        <!-- <ul class="nav navbar-nav"> -->

        <!-- <li><a class="nav-link"   [routerLink]="['/home']"><b>Home</b></a></li> --
>
          <!-- <li><a class ="nav-link" [routerLink]="['/order']"><b>Order</b></a></li> --
>
          <!-- <li><a class ="nav-link" [routerLink]="['/login']"><b>Login</b></a></li> --
>
          <!-- <li ><a class ="nav-link" [routerLink]="['/about']"><b>About</b></a></li>
-->
          <!-- <li ><a href="#contact"><b>Contact</b></a></li> -->

        <!-- </ul> -->

       <!-- </div> -->
     <!-- </div> -->
   <!-- </nav> -->
<html>

<head>
<style>
  * {
    font-family: sans-serif;
    font-size: 9pt;
}
nav {
    padding: 0;
```

```
    font-size:15;
}
.container {
    width: 85%;
    margin: 30px auto 0 auto;
    position: relative;

}
#links {
    padding: 0;
    overflow: auto;
    margin: 0 auto;
    list-style: none;
    width: 100%;
    margin: 0 auto;
    font-size: 15;
}
li {
    list-style-type: none;
    float: left;
    width: 20%;
    padding: 0;
    margin: 0;
}
#links li a {
    text-decoration: none;
    text-transform: uppercase;
    color: black;
    display: block;
    text-align: center;
    padding: 8px 0;
    font-size: 15;
}
.progressBar {
    height: 5px;
    overflow: hidden;
    background-color: lightgray;
    left: 0;
    right: 0;
}
#progress {
    width: 20%;
    background-color: red;
    height: 100%;
    transition: width 0.5s;
}
```

```
.active a {
   background-color: black;
   color: white !important;
   transition: background-color 0.5s;
}

 </style>
 <script>
 var progress = document.getElementById("progress"),
   navBar = document.getElementById("links"),
   linkCount = document.getElementsByTagName("li"),
   oneLink = navBar.offsetWidth / linkCount.length;

progress.offsetWidth = Math.ceil(oneLink) + "px";
linkCount[0].classList.add("active");

function red(pos, link) {
   for (var i = 0; i < linkCount.length; i++) {
      linkCount[i].classList.remove("active");
   }
   progress.style.width = oneLink * pos + "px";
   link.classList.add("active");
}

// RELOAD ON RESIZE
window.onresize = function() {
   location.reload();
};

 </script>
</head>
<body>
   <div class="navbar-header">
      <!-- <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button> -->
    <a class="navbar-brand" href="index.html"><img src="../../assets/Swito.jpg" margin-
top="6px" vspace="1"></a>
     </div>

   <div class="container">
     <nav>
        <ul id="links">
```

```
          <li onclick="red(1, linkCount[0])">
            <a class="nav-link"   [routerLink]="['/home']"><b><h2>'</h2></b></a>
          </li>
          <li onclick="red(1, linkCount[1])">
             <a class="nav-link"   [routerLink]="['/home']"><b>Home</b></a>
            </li>
          <li onclick="red(2, linkCount[1])">
            <a class ="nav-link" [routerLink]="['/cart']"><b>Order</b></a>
          </li>
          <!-- <div class="dropdown">
            <a [routerLink]="['/login']" class="nav-link">Login</a>
              <div class="dropdown-content">
                <a [routerLink]="['/login']">Canteen Login</a>
                <a [routerLink]="['/viewitems']">Developer Login</a>


              </div>
            </div> -->
          <li onclick="red(3, linkCount[2])">
            <a class ="nav-link" [routerLink]="['/login']"><b>Login</b></a>
          </li>
          <li onclick="red(4, linkCount[3])">
            <a class ="nav-link" [routerLink]="['/about']"><b>About</b></a>
          </li>
          <!-- <li onclick="red(4, linkCount[3])">
            <a class ="nav-link" [routerLink]="['/join']"><b>Join</b></a>
          </li> -->

        </ul>
        <div class="progressBar">
           <div id="progress"></div>
        </div>
      </nav>
    </div>
    </body>
```

footer.component.html

```
<div id="contact">
   <div class="container">
     <div class="text-center">
       <h3>Contact Us</h3>
       <p>get best snacks on your finger tips</p>
     </div>
   </div>
   <div class="container">
     <div class="row">
       <div class="col-md-4 wow fadeInUp" data-wow-offset="0" data-wow-delay="0.2s">
         <h2>Contact us any time</h2>
         <p>please send us your details along with your query our team will contact you ,keep ordering with best served canteens in the country.</p>
       </div>

       <div class="col-md-4 wow fadeInUp" data-wow-offset="0" data-wow-delay="0.4s">
         <h2>Contact Info</h2>
         <ul>
           <li><i class="fa fa-home fa-2x"></i> iitm janakpuri , d block, new delhi,india</li>
           <hr>
           <li><i class="fa fa-phone fa-2x"></i> +910298786758</li>
           <hr>
           <li><i class="fa fa-envelope fa-2x"></i> info@swito.com</li>
         </ul>
       </div>

       <div class="col-md-4 wow fadeInUp" data-wow-offset="0" data-wow-delay="0.6s">
         <div id="sendmessage">Your message has been sent. Thank you!</div>
         <div id="errormessage"></div>
         <form action="" name="myform" method="post" role="form" class="contactForm" #regForm='ngForm' (ngSubmit)="Register(refForm)">


           <div class="form-group">
             <input type="text" name="name" class="form-control" id="name" placeholder="Your Name" [(ngModel)]="name" name="name" data-rule="minlen:4" ng-minlength=3 message required />
             <!-- <div *ngIf="name.invalid && (name.dirty || name.touched)" class="alert alert-danger">
```

```
            <div *ngIf="name.errors.required">
              Name is required.
             </div>
             <div *ngIf="name.errors.minlength">
              Name must be at least 4 characters long.
             </div>
             <div *ngIf="name.errors.forbiddenName">
              Name cannot be Bob.
             </div>

            </div> -->

          <!-- <span ng-show="myform.name.$touched && myform.name.$error.required">First name is required.</span><br /><br /> -->
          <!-- <span class="help-block" ng-show="errorFName">{{ errorFname }}</span> -->
          <!-- <span ng-show="myform.name.$touched && myform.name.$invalid">The name is required.</span> -->
        </div>
        <div class="form-group">
          <input type="email" class="form-control" id="email" placeholder="Your Email" data-rule="email" [(ngModel)]="email" name="email" data-msg="Please enter a valid email" email required />
          <span class="help-bpx" *ngIf="email.touched && !email.valid">Please enter valid email</span>
          <div class="validation"></div>
        </div>
        <div class="form-group">
          <input type="text" class="form-control" name="subject" id="subject" placeholder="Subject" data-rule="minlen:4" [(ngModel)]="subject" name="subject" data-msg="Please enter at least 8 chars of subject" subject required/>


          <div class="validation"></div>
        </div>
        <div class="form-group">
          <textarea class="form-control" name="message" rows="5" data-rule="required" data-msg="Please write something for us" placeholder="message" [(ngModel)]="message" ng-minlength=10 message required></textarea>
          <div class="validation"></div>
        </div>

        <button type="submit" class="btn btn-theme pull-left" (click)="addData()" [disabled]="!regForm.valid">SEND MESSAGE</button>
      </form>
```

```
      </div>
     </div>
    </div>
  </div>
  <!--/#contact-->
  <!-- <div class="container">
    <div class="sub-footer">
     <div class="text-center">
      <div class="col-md-12">
       <form class="form-inline">
        <div class="form-group">
         <button type="purchase" name="purchase" class="btn btn-primary btn-lg" required="required">Enter Your Email</button>
        </div>
        <div class="form-group">
         <button type="subscribe" name="subscribe" class="btn btn-primary btn-lg" required="required">Subscribe Now</button>
        </div>
       </form>
      </div>
     </div>
    </div>
  </div>-->
  <div class="social-icon">
    <div class="container">
     <div class="col-md-6 col-md-offset-3">
      <ul class="social-network">
       <li><a href="https://www.facebook.com/jainarayan.jha" class="fb tool-tip" title="Facebook"><i class="fa fa-facebook"></i></a></li>
       <li><a href="https://www.instagram.com/rahul_choudhary7210/?hl=en" class="instagram tool-tip" title="Twitter"><i class="fa fa-twitter"></i></a></li>
       <li><a href="https://twitter.com/Jayjha10?s=08" class="gplus tool-tip" title="Google Plus"><i class="fa fa-google-plus"></i></a></li>
       </ul>
     </div>
    </div>
  </div>
   <!-- <div class="text-center">
    <button ><a href="#main-navbar"> go to top</a></button>
  </div> -->
```

adminorders.component.html

```
<app-admindashboard> </app-admindashboard>
<app-header2> </app-header2>
<html>
   <head>
      <style>
.raa{background-color:rgb(44, 3, 3);color: blanchedalmond;font-family:cursive;}

      </style>
   </head>
   <body>
      <div class=raa>

      <!-- {{obj.canteenname}}       -->
      <div id="nav-bar" *ngFor="let obj of arr">
      <h1> Welcome  {{obj.canteenname}}</h1>
      </div>
      </div>
         <!-- {{obj.canteenname}} -->
         <!-- {{obj.canteenname}} -->
<img src="../../assets/portfolio/9.jpg" height=650 width=1600/>

   </body>
</html>
```

admindashboard.component.html

```
<html>
  <head>
    <style>
.row, .article-left{
  padding: 0;
  margin: 0;
}

.article-right{
  background-color: #FFF;
  padding-top: 30px;
}

ul li{
  text-transform: Uppercase;
  font-size: 1.8em;
  padding: 15px;
}
```

```
ul li a{
  color: #FFF;
}

a:hover{
  color: #DDD;
  text-decoration: none;
}

h1{
  font-size:3em;
  color: #333;
  padding-top: 15px;
}

h2{
   background-color: rgb(25, 34, 43);
   padding: 15px;
}

.title{
  background-color: rgb(44,64,83);
  padding: 10px;
}
ul li, .fa, h2, h3{
  color: #FFF;
}
.fa{
  padding-right: 20px;
}
.fa-heart-o{
  padding-left: 150px;
  padding-top: 20px;
  padding-bottom: 30px;
}
    </style>
  </head>
  <body> <br><br> <br>
    <br>
    <h2>Admin Dashboard</h2>
  </body>
</html>
```

aboutus.component.html

```
<div class="table">
```

Swito is a canteen management website which is made by Rahul Ranjan and Jai jha. Both are the pursuing BCA from Institute of Innovation in Techology and management which is affiliated to Guru Gobind Singh Indraprashtha University.

As we have traditional canteen management system,so this website aim to provide digital way to eliminate problems of traditional canteen sysytem.

The problems we face in canteen are standing in queues and it takes lot of time to give orders and thus canteen management also requires a lot of more human resource to run it.

This application has two interfaces: one for client side and the other for server side. The client side mainly focuses on customers who can place their orders and doesn't require any credentials and thus can make payment through their phones on the spot. Once an order is placed a token is generated automatically.

On the server side, we provide a user interface for admin or the owners of the canteen who can log in to this application and create their menu and add payment methods. This application supports multiple canteens and a lot of canteens can register. Admins are responsible to update, verify tokens and accept or to reject orders.

```
<app-header></app-header>
<section id="about" class="section">
   <div class="container">
     <div class="row">
       <div class="col-md-8 col-md-offset-2">
         <div class="heading">
           <h2 >      About us</h2>
         </div>
         <div class="sub-heading">
           <p>
                 </p>
         </div>
       </div>
     </div>
<!-- new -->
<app-aboutus></app-aboutus>

  <!-- </section> -->
  <app-raa4></app-raa4>
  <!-- <app-validation></app-validation> -->
```

```
<app-validation></app-validation>
  <!-- <app-cart></app-cart> -->
  <!-- <app-cartlogo></app-cartlogo> -->
  <app-footer></app-footer>
```

forgotpaw.component.html

```
<app-header></app-header>
<html>
    <head>
        <style>
            body {
  font-family: 'Raleway', sans-serif;
  font-weight: 300;
  color:#fff;
  line-height:1.2;
  margin:0;
  background-image: url("../../assets/sand.jpg");
  background-repeat:repeat;
  background-size:cover;
  font-size: 18px;
}
.info {
  margin-bottom:50px;
}
.info p {
  text-align:center;
  color: #fff;
  text-transform:none;
  font-weight:500;
  font-size:17px;
  margin-top:2px
}

.info i {
  color:red;
}
.clearfix {clear:both;}

textarea:focus, input:focus{
    outline: none;
}

h1 {
  text-align:center;
  color: #fff;
```

```
  text-shadow: 1px 1px 0px #222;
  margin:50px 0px 0px 0px;
  font-size:32px;
  font-weight:500;
}
h2 {
  font-weight:300;
  font-size:27px;
  margin:0px;
  text-align:center;
  text-transform:uppercase;
  margin-top:50px;
  padding-bottom:15px;
}
a {
  text-decoration: inherit;
  color: inherit;
}

.container {
  width:560px;
  height:530px;
  margin:auto;
  display:block;
  background-color:rgba(0,0,0,0.5);
  border-radius:15px;
  padding:15px;
  margin-bottom:50px;
}
.login-form {
  position:relative;
  border:2px solid rgba(255,255,255,0.2);
  width:99%;
  height:99%;
  margin:auto;
  display:block;
  /* vertical-align:middle; */
  border-radius:15px;
}
.login-form hr.to-right {
  width: 45%;
  position:absolute;
  right:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to left,transparent, #F9E206);
```

```
}
.login-form hr.to-left {
  width: 45%;
  position:absolute;
  left:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to right,transparent, #F9E206);
}

.container-login {
  width:60%;
  margin:auto;
  display:block;
  margin-top:30px;
}

label {
  width: 100%;
  display: inline-block;
  padding:0px 0px 5px 5px;
  margin-top: 20px;
  font-size:18px;
}
input {
  border: 0px none;
  width: 94%;
  border-radius: 4px;
  height: 25px;
  padding: 5px 3%;
  font-size:15px;
  font-family: inherit;
  color:#222;
}
button {
        width:100%;
        background-color:rgba(74,129,51,0.4);
        border:2px solid #6FC149;
        color:#fff;
        border-radius:5px;
        height:50px;
        text-transform:uppercase;
        font-size:19px;
        cursor:pointer;
        transition:all 0.3s;
}
```

```
.login-footer {
	margin-top:50px;
}
.login-footer a {
	font-size: 16px;
	margin:5px 0px;
	text-align:center;
	color:rgba(255,255,255,0.4);
	font-weight:400;
	line-height:1.4;
	display: block;
	transition:all 0.2s;
}
.login-footer a:hover {
	color:rgba(255,255,255,0.7);
}
.login-footer i {
	padding:0px 6px 0px 0px;
}

.check {
	padding: 20px 0px;
}
.check label {
   display:inline;
	/* margin-top:5px; */
	position: relative;
	top: 5px;
}
.check p {
   display:inline;
	margin-left: 8px;
}
.checkbox {
   display: none;
}

.checkbox:checked + svg .path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   stroke-dasharray: 25 90;
   stroke-dashoffset: 0;
}
```

```
.path-moving,
.path-back {
   fill: none;
   stroke: #64A546;
   stroke-width: 3px;
   stroke-linecap: round;
   stroke-linejoin: round;
}

.path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   stroke: #ffffff;
   stroke-dasharray: 110;
   stroke-dashoffset: -32;
}
        </style>

        <!-- <link href="assets/css/login.css" rel="stylesheet" /> -->
   </head>
<body>

      <div class="info"><a href="#" target="_blank"><p>  <i class="fa fa-heart"></i>
</p></a></div>

      <div class="container">
        <div class="login-form">

          <h2>Reset Password</h2>
          <hr class="to-right">
          <hr class="to-left">

             <div class="container-login">

                <label>Email Id</label>
                <input type="text" class="form-control" id="inputEmail4"
[(ngModel)]="email" placeholder="your email"/>
                <p>Enter your email above and we'll send you the Reset Link.</p>

                 <button type="submit" class="btn btn-primary" (click)="signIn2()" > <a
[routerLink]="[ '/login/forgot password/resetpw' ]">confirm</a> </button>
                 <!-- <a [routerLink]="[ '/login/forgot password/resetpw' ]">confirm</a> --
>
             </div>
```

```
        </div>
      </div>
    </body>
    </html>
    <script>
    const addName = document.querySelector('.name')
const addSkills = document.querySelector('.skills')
const createBtn = document.querySelector('button')
const cardContainer = document.querySelector('section')

let users = []

createBtn.addEventListener('click', newCard)

function newCard () {

  const card =  `<div class="card">
  <h3>${addName.value}</h3>
  <ul>${addSkills.value.split(',').map(value =>
  `<li>${value.trim()}</li>`).join('')}</ul>`


  addName.value=""
  addSkills.value=""
  addName.focus()

 cardContainer.insertAdjacentHTML('afterbegin', card)
}
</script>
<app-footer></app-footer>
```

login.component.html

```
<app-header></app-header>
<html>
   <head>
      <style>
         body {
  font-family: 'Raleway', sans-serif;
  font-weight: 300;
  color:#fff;
  line-height:1.2;
  margin:0;
  background-image: url("../../assets/login.jpg");
  background-repeat:repeat;
  background-size:cover;
  font-size: 18px;
}
.info {
  margin-bottom:50px;
}
.info p {
  text-align:center;
  color: #fff;
  text-transform:none;
  font-weight:500;
  font-size:17px;
  margin-top:2px
}

.info i {
  color:red;
}
.clearfix {clear:both;}

textarea:focus, input:focus{
    outline: none;
}

h1 {
  text-align:center;
  color: #fff;
  text-shadow: 1px 1px 0px #222;
  margin:50px 0px 0px 0px;
  font-size:32px;
  font-weight:500;
```

```
}
h2 {
  font-weight:300;
  font-size:27px;
  margin:0px;
  text-align:center;
  text-transform:uppercase;
  margin-top:50px;
  padding-bottom:15px;
}
a {
  text-decoration: inherit;
  color: inherit;
}

.container {
  width:560px;
  height:530px;
  margin:auto;
  display:block;
  background-color:rgba(0,0,0,0.5);
  border-radius:15px;
  padding:15px;
  margin-bottom:50px;
}
.login-form {
  position:relative;
  border:2px solid rgba(255,255,255,0.2);
  width:99%;
  height:99%;
  margin:auto;
  display:block;
  /* vertical-align:middle; */
  border-radius:15px;
}
.login-form hr.to-right {
  width: 45%;
  position:absolute;
  right:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to left,transparent, #F9E206);
}
.login-form hr.to-left {
  width: 45%;
  position:absolute;
```

```
  left:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to right,transparent, #F9E206);
}

.container-login {
  width:60%;
  margin:auto;
  display:block;
  margin-top:30px;
}

label {
  width: 100%;
  display: inline-block;
  padding:0px 0px 5px 5px;
  margin-top: 20px;
  font-size:18px;
}
input {
  border: 0px none;
  width: 94%;
  border-radius: 4px;
  height: 25px;
  padding: 5px 3%;
  font-size:15px;
  font-family: inherit;
  color:#222;
}
button {
        width:100%;
        background-color:rgba(74,129,51,0.4);
        border:2px solid #6FC149;
        color:#fff;
        border-radius:5px;
        height:50px;
        text-transform:uppercase;
        font-size:19px;
        cursor:pointer;
        transition:all 0.3s;
}

.login-footer {
        margin-top:50px;
}
```

```
.login-footer a {
	font-size: 16px;
	margin:5px 0px;
	text-align:center;
	color:rgba(255,255,255,0.4);
	font-weight:400;
	line-height:1.4;
	display: block;
	transition:all 0.2s;
}
.login-footer a:hover {
	color:rgba(255,255,255,0.7);
}
.login-footer i {
	padding:0px 6px 0px 0px;
}

.check {
	padding: 20px 0px;
}
.check label {
   display:inline;
	/* margin-top:5px; */
	position: relative;
	top: 5px;
}
.check p {
   display:inline;
	margin-left: 8px;
}
.checkbox {
   display: none;
}

.checkbox:checked + svg .path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   stroke-dasharray: 25 90;
   stroke-dashoffset: 0;
}

.path-moving,
.path-back {
   fill: none;
```

```
   stroke: #64A546;
   stroke-width: 3px;
   stroke-linecap: round;
   stroke-linejoin: round;
}

.path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   stroke: #ffffff;
   stroke-dasharray: 110;
   stroke-dashoffset: -32;
}
         </style>
         <!-- <link href="assets/css/login.css" rel="stylesheet" /> -->
   </head>
<body>

      <div class="info"><a href="#" target="_blank"><p>  <i class="fa fa-heart"></i>
</p></a></div>

      <div class="container">
        <div class="login-form">

           <h2>Welcome to Swito</h2>
           <hr class="to-right">
           <hr class="to-left">

              <div class="container-login">

                 <label>Username</label>
                 <input type="text" class="form-control" [(ngModel)]="email"
name="email" placeholder="your name"/>

                 <label>Password</label>
                 <input type="password" class="form-control" id="inputPassword4"
[(ngModel)]="password"  placeholder="●●●●●●●●●●" />

                 <div class="check">
                       <label>
                          <input id="check" type="checkbox" class="checkbox">
                             <svg xmlns="http://www.w3.org/2000/svg" width="26px"
height="23px">
                                 <path class="path-back"
d="M1.5,6.021V2.451C1.5,2.009,1.646,1.5,2.3,1.5h18.4c0.442,0,0.8,0.358,0.8,0.801v18.
398c0,0.442-0.357,0.801-0.8,0.801H2.3c-0.442,0-0.8-0.358-0.8-0.801V6"/>
```

```
                                <path class="path-moving"
d="M24.192,3.813L11.818,16.188L1.5,6.021V2.451C1.5,2.009,1.646,1.5,2.3,1.5h18.4c0
.442,0,0.8,0.358,0.8,0.801v18.398c0,0.442-0.357,0.801-0.8,0.801H2.3c-0.442,0-0.8-
0.358-0.8-0.801V6"/>
                            </svg>
                        </label>
                        <p>Keep me signed in</p>
                    </div>
                     <!-- <a [routerLink]="['/adminorders']">  -->
                      <button type="submit" class="btn btn-primary"
(click)="signIn()">Login !</button>
                    <!-- </a> -->


                <div class="login-footer">
                    <a [routerLink]="[ '.', 'forgot password' ]"><i class="fa fa-exclamation-
circle"></i> Forgot your password ?</a>
                    <a [routerLink]="[ '.', 'sign up' ]"><i class="fa fa-thumbs-up"></i>
Create an account </a>
                </div>
            </div>

        </div>
      </div>

    </body>
    <app-footer></app-footer>

    </html>
```

signup.component.html

```
<app-header></app-header>
<html>
   <head>
      <style>
         body {
  font-family: 'Raleway', sans-serif;
  font-weight: 300;
  color:#fff;
  line-height:1.2;
  margin:0;
  background-image: url("../../assets/sand.jpg");
  background-repeat:repeat;
  background-size:cover;
  font-size: 18px;
}
.info {
  margin-bottom:50px;
}
.info p {
  text-align:center;
  color: #fff;
  text-transform:none;
  font-weight:500;
  font-size:17px;
  margin-top:2px
}

.info i {
  color:red;
}
.clearfix {clear:both;}

textarea:focus, input:focus{
    outline: none;
}

h1 {
  text-align:center;
  color: #fff;
  text-shadow: 1px 1px 0px #222;
  margin:50px 0px 0px 0px;
  font-size:32px;
  font-weight:500;
```

```
}
h2 {
  font-weight:300;
  font-size:27px;
  margin:0px;
  text-align:center;
  text-transform:uppercase;
  margin-top:50px;
  padding-bottom:15px;
}
a {
  text-decoration: inherit;
  color: inherit;
}

.container {
  width:560px;
  height:730px;
  margin:auto;
  display:block;
  background-color:rgba(0,0,0,0.5);
  border-radius:15px;
  padding:15px;
  margin-bottom:50px;
}
.login-form {
  position:relative;
  border:2px solid rgba(255,255,255,0.2);
  width:99%;
  height:99%;
  margin:auto;
  display:block;
  vertical-align:middle;
  border-radius:15px;
}
.login-form hr.to-right {
  width: 45%;
  position:absolute;
  right:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to left,transparent, #F9E206);
}
.login-form hr.to-left {
  width: 45%;
  position:absolute;
```

```
  left:30px;
  border: 0px none;
  height: 3px;
  background-image: linear-gradient(to right,transparent, #F9E206);
}

.container-login {
  width:60%;
  margin:auto;
  display:block;
  margin-top:30px;
}
textarea{width: 100%;
  display: inline-block;
  padding:0px 0px 5px 5px;
  margin-top: 20px;
  font-size:18px;}

label {
  width: 100%;
  display: inline-block;
  padding:0px 0px 5px 5px;
  margin-top: 20px;
  font-size:18px;
}
input {
  border: 0px none;
  width: 94%;
  border-radius: 4px;
  height: 25px;
  padding: 5px 3%;
  font-size:15px;
  font-family: inherit;
  color:#222;
}
button {
        width:100%;
        background-color:rgba(74,129,51,0.4);
        border:2px solid #6FC149;
        color:#fff;
        border-radius:5px;
        height:50px;
        text-transform:uppercase;
        font-size:19px;
        cursor:pointer;
        transition:all 0.3s;
```

```
  margin-top: 20px;
}

.login-footer {
        margin-top:50px;
}
.login-footer a {
        font-size: 16px;
        margin:5px 0px;
        text-align:center;
        color:rgba(255,255,255,0.4);
        font-weight:400;
        line-height:1.4;
        display: block;
        transition:all 0.2s;
}
.login-footer a:hover {
        color:rgba(255,255,255,0.7);
}
.login-footer i {
        padding:0px 6px 0px 0px;
}

.check {
        padding: 20px 0px;
}
.check label {
   display:inline;
        margin-top:5px;
        position: relative;
        top: 5px;
}
.check p {
   display:inline;
        margin-left: 8px;
}
.checkbox {
   display: none;
}

.checkbox:checked + svg .path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s cubic-
bezier(.3,.8,.6,1.5);
   stroke-dasharray: 25 90;
```

```
   stroke-dashoffset: 0;
}

.path-moving,
.path-back {
   fill: none;
   stroke: #64A546;
   stroke-width: 3px;
   stroke-linecap: round;
   stroke-linejoin: round;
}

.path-moving {
   -webkit-transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   transition: stroke .4s,stroke-dasharray .4s,stroke-dashoffset .4s;
   stroke: #ffffff;
   stroke-dasharray: 110;
   stroke-dashoffset: -32;
}
         </style>
   </head>
<body>

      <div class="info"><a href="#" target="_blank"><p>  <i class="fa fa-heart"></i>
</p></a></div>

      <div class="container">
        <div class="login-form">

           <h2>enter your details</h2>
           <hr class="to-right">
           <hr class="to-left">

              <div class="container-login">

                 <label>name</label>
                 <input type="text" type="text" class="form-control" [(ngModel)]="name"
name="name" placeholder="your name"/>

                 <label>email</label>
                 <input type="text" type="email" class="form-control" id="inputEmail4"
[(ngModel)]="email"  name="email" placeholder="●●●●●●●●●●" />
                 <label>canteen name</label>
                 <input type="text" class="form-control" [(ngModel)]="canteenname"
name="canteenname" placeholder="canteen name" />
```

```
                <label>phone</label>
                <input type="number"  class="form-control" [(ngModel)]="phone"
name="phone" placeholder="enter phone number"/>
                <label>password</label>
                <input type="text" type="password" class="form-control"
id="inputPassword4" [(ngModel)]="password" placeholder="●●●●●●●●●●" />

                <label>Address</label>
                <textarea class="form-control" [(ngModel)]="address" name="address"
></textarea>


                <a [routerLink]="['/login']">  <button type="submit" class="btn btn-
primary" (click)="addData()" >Create Account</button></a>


            </div>

        </div>
      </div>

    </body>

    </html>
    <app-footer></app-footer>
```

logout.component.html

```
<!-- <app-admindashboard> </app-admindashboard> -->
<app-header> </app-header>
<html>
   <head>
      <style>

@import url(https://fonts.googleapis.com/css?family=Open+Sans:300,400);
@import url(https://fonts.googleapis.com/css?family=Mr+Dafoe:300,400);

html {
  background: #010527 center/cover no-repeat fixed;
/*  background: #4889BF center/cover no-repeat fixed;*/
}
body{
   background:black;
}
.sign {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%) rotate(-45deg) skew(20deg, 20deg);
  width: 350px;
  height: 350px;
  background: #EAE7DF;
  box-shadow: inset 0 0 0 10px #ffCB3A;
  border: 5px dotted #E6D1b2;
  border-top-left-radius: 70px;
  border-bottom-right-radius: 70px;
}
.text {
  position: absolute;
  top: 49%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
  white-space: nowrap;
  font-family: Helvetica, sans-serif;
  color: #0261DF;
}
.welcome {
  display: flex;
}
.welcome span {
```

```
  display: inline-block;
  width: 32px;
  height: 30px;
  padding: 15px;
  font-size: 30px;
  font-weight: bold;
  color: #FF020D;
  background: #EAE7DF;
  border: 1px solid #DEC08C;
  border-radius: 50%;
  box-shadow: 0px 10px 5px #5D646E;
}
.text .to {
  position: relative;
  top: 28px;
  left: -140px;
  family: 'Open Sans';
  font-size: 26px;
  font-weight: 400;
}
.text .fab {
  position: relative;
  top: -14px;
  left: 5px;
  font-family: 'Mr Dafoe';
  font-size: 64px;
  line-height: 50px;
  letter-spacing:5px;
}
.text .las {
  position: relative;
  top: -20px;
  left: 8px;
  font-size: 66px;
  font-weight: 600;
  letter-spacing: 5px;
  color: red;
}
.text .nv {
  position: relative;
  top: -25px;
  left: 0;
  font-size: 30px;
  font-weight: 600;
  letter-spacing: 6px;
}
```

```
        /* </style>
        <script type="text/javascript">
function preback(){window.history.forward();}
setTimeout("preback()",0);
window.onunload=function(){null};
        </script> */

<!-- (function (global) {

if(typeof (global) === "undefined") {
    throw new Error("window is undefined");
}

var _hash = "!";
var noBackPlease = function () {
    global.location.href += "#";

    // making sure we have the fruit available for juice (^__^)
    global.setTimeout(function () {
        global.location.href += "!";
    }, 50);
};

global.onhashchange = function () {
    if (global.location.hash !== _hash) {
        global.location.hash = _hash;
    }
};

global.onload = function () {
    noBackPlease();

    // disables backspace on page except on input fields and textarea..
    document.body.onkeydown = function (e) {
        var elm = e.target.nodeName.toLowerCase();
        if (e.which === 8 && (elm !== 'input' && elm  !== 'textarea')) {
            e.preventDefault();
        }
        // stopping event bubbling up the DOM tree..
        e.stopPropagation();
    };
}

})(window);
        </script> -->
```

```
  </head>
  <body>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <main>
      <div class="sign"></div>
      <div class="text">

        <div class="to">

        </div>
        <div class="fab">You have been logged out</div>
        <div class="las">Thanks!!</div>
        <div class="nv">for using us</div>

      </div>
    </main>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
  </body>
</html>
<app-footer></app-footer>
```

tokenid.component.html

```
<html>
   <head>
      <script>
         $( document ).ready(function() {

// set the length of the string
var stringLength = 5;

// list containing characters for the random string
var stringArray =
['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f','g','h','i','j','v','w','x','y','z','A','B','C','D','E','F','G
','V','W','X','Y','Z','!','?'];

$("#generateToken").click(function (){

   var rndString = "";

   // build a string with random characters
   for (var i = 1; i < 5; i++) {
      var rndNum = Math.ceil(Math.random() * stringArray.length) - 1;
      rndString = rndString + stringArray[rndNum];
   };

   $("#showToken").html('<p><strong>' + rndString + '<strong></p>');
alert(hello);
});

});
         </script>
   </head>
<body>

      <p>Click the button below to generate a  token.</p>
      <button id="generateToken">Generate New Token</button>
      <p id="showToken"></p>

</body>
</html>
```

view.component.html

```
<div class="row">
   <div class="col-3" *ngFor="let obj of arr">
      <div class="card" style="width: 18rem; "  >
            <img class="card-img-top"
             src="{{obj.imageLink}}" alt="Card image cap">
            <div class="card-body">
               <h5 class="card-title">Name : {{obj.name}}</h5>
               <p class="card-text">Email : {{obj.email}}</p>
               <p class="card-text">Height : {{obj.height}}</p>
               <p class="card-text">Weight : {{obj.weight}}</p>
               <a href="#" class="btn btn-primary">
                  {{obj.password}}</a>

       <button (click)="delData(obj._id)">Delete</button>
            </div>
         </div>
         <!-- <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
aria-labelledby="{{obj._id}}" aria-hidden="true">
               <div class="modal-dialog" role="document">
                 <div class="modal-content">
                   <div class="modal-header">
                     <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
                     <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                       <span aria-hidden="true">&times;</span>
                     </button>
                   </div>
                   <div class="modal-body">
                     ...
                   </div>
                   <div class="modal-footer">
                     <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
                     <button type="button" class="btn btn-primary">Save changes</button>
                   </div>
                 </div>
               </div>
             </div> -->
      </div>
```

welcome.component.html

```
<html>
  <head>
    <style>
    @import url(https://fonts.googleapis.com/css?family=Fjalla+One);
@import url(https://fonts.googleapis.com/css?family=Oswald);
@import url(https://fonts.googleapis.com/css?family=Montserrat);

html, body {
  /* https://i.imgur.com/cKUIHOz.png */
 /* background: url('https://i.imgur.com/Pxa0p0m.jpg') center fixed;*/
  background-size: cover;
  overflow-x: hidden;
}

#nav-bar {
  background-color: #1c1c1f;
  position: absolute;
  top: 0;
  left: 0;
  height: 110px;
  width: 964px;
  border-radius: 0px 0px 90px 0px;
  box-shadow: 0px 0px 7px #000;
}

#nav-bar-container {
  margin-top: 24px;
  display: block;
}

.nav-bar-item {
  font-family: 'Montserrat', sans-serif;
  font-size: 23px;
  text-decoration: none;
  color: #fff;
  padding: 5px;
  margin-left: 55px;
  margin-right: 0px;
  float: left;
  transition: 0.2s ease-in-out;
  letter-spacing: -1px;
}
```

```
.nav-bar-item:hover {
  color: #DBAD2E;
}

#site-title-icon {
  float: left;
  height: 60px;
  margin-top: 14px;
  margin-left: 17px;
}

#big-text {
  font-family: 'Fjalla One', sans-serif;
  font-size: 90px;
  color: #F5C43C;
  position: absolute;
  top: 30.5%;
  left: 54px;
  /* font-smoothing: antialiased; */
  letter-spacing: -1.1px;
  text-transform: uppercase;
  min-width: 700px;
}

#big-text-sub {
  font-family: 'Montserrat', sans-serif;
  font-size: 25px;
  color: #DBAD2E;
  position: absolute;
  top: 30.5%;
  left: 55px;
  margin-top: 176px;
  /* font-smoothing: antialiased; */
  letter-spacing: -1.1px;
  text-transform: uppercase;
  min-width: 710px;
}

#footer {
  background-color: #1c1c1f;
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  height: 55px;
  min-width: 900px;
```

```
}

#site-copyright {
  font-family: 'Montserrat', sans-serif;
  font-weight: 100;
  font-size: 16px;
  color: #515151;
  position: absolute;
  bottom: 0;
  right: 23px;
  top: 3px;
}

#social-icon-youtube {
  height: 45px;
  position: absolute;
  top: 6px;
  left: 60px;
}

#social-icon-youtube:hover {
  cursor: pointer;
}

#social-icon-twitter {
  height: 45px;
  position: absolute;
  top: 6px;
  left: 9px;
}

#social-icon-twitter:hover {
  cursor: pointer;
}

#side-bar {
  visibility: hidden;
}

#side-bar-item {
  visibility: hidden;
}

#side-bar-icon {
  visibility: hidden;
}
```

```
@media(max-height: 600px) {
  #footer {
    display: none;
  }

  #big-text {
    font-size: 55px;
    top: 20.5%;
    left: 54px;
    margin-top: 112px;
  }

  #big-text-sub {
    top: 20.5%;
    left: 55px;
  }
}
.dropdown {
  position: relative;
  display: inline-block;
}
.dropdown-content {
  display: none;
  position: absolute;
  padding: 15px;
  background-color: #1c1c1f;
  top: 100%;
  border-bottom-left-radius: 2px;
  border-bottom-right-radius: 2px;
}
.dropdown-content a {
  display: inline-block;
  padding-bottom: 18px;
}
.dropdown-content a:last-child {
  padding-bottom: 0px;
}
.dropdown:hover .dropdown-content {
  display: block;
}
a:hover {
  color: white;
  text-decoration: underline;
}
#ra{
```

```
   text-align: center;
   color:white;
}
   </style>
<script>
var $el = $('#big-text'),
   txt = $el.text(),
   txtLen = txt.length,
   timeOut,
   char = 0;

$el.text('|');

(function typeIt() {
   var humanize = Math.round(Math.random() * (200 - 30)) + 50;

   timeOut = setTimeout(function() {
      char++;
      var type = txt.substring(0, char);
      $el.text(type + '');
      typeIt();

      if (char == txtLen) {
         $el.text($el.text().slice(0, txt.length));
         clearTimeout(timeOut);
      }
   }, humanize);
}());
      </script>
</head>
      <body>
  <br>
  <br>
  <br>
  <!-- {{obj.canteenname}}       -->
                  <div id="nav-bar" *ngFor="let obj of arr">


                    <div id="ra">
                      <h1>Hola Customer!!! - {{obj.canteenname}}   </h1>
                      <h3> You are Important to us -  {{obj.name}}  </h3>
                      </div>
                      </div>

</body>
</html>
```

Chapter – 5 Scope of Improvements and Conclusions

5.1 Scope of Improvement

Swito, a Canteen Management App and Website is for small owners and managers of cafeteria,it digitalizes the present system of canteen management and takes it to a whole new level. It offers owners of canteens and small cafeteria to take their business online, this project containsboth website and mobile application, in our website the canteen owners can register their canteen and soon after successful registration they will get access to admin panel from wherethey will be able to upload all the required details which will be reviewed by our team and then it will be made available for the customers, owners can only register or login to admin panel through our website.

Company Canteen Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

5.2 System Summary

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable ata/information can be stored for a longer period with easy accessing and manipulation of thesame. Basically, the project describes how to manage for good performance and better servicesfor the clients.

The main objective of the Project on Canteen Management System is to book the snack onlineand pay online. It manages all the information about Bill Payment. The purpose of the projectis to build an application program to reduce the manual work for managing the Bill Payment,snack ordering. It tracks all the details about the customer Meal, Meal Type, Canteen menu. Swito, a Canteen Management App and Website is for small owners and managers of cafeteria,it digitalizes the present system of canteen management and takes it to a whole new level. Itoffers owners of canteens and small cafeteria to take their business online, The problems we face in canteen are standing in queues and it takes lot of time to orders and thus canteen management also requires a lot of more human resource to run it.

The benefit of this is that if there is rush in the Canteen then there will be chances that the waiters will be unavailable and the users can directly order the food to the chef online by usingthis application.

This Canteen Automation System enables the end users to register online, read and select thefood from e-menu card and order food online by just selecting the food that the user want to have using android application. The results after selecting the food from the E-menu card willdirectly appear in the screen near the Chef who is going to cook the food for you.

In future SWITO can be used in MNC’s and also in colleges for canteen, this will save the time of customer as well as the workers of canteen management.

5.3 Limitations

In current scenario there are cashiers in canteen so we have to educate them so they will be able to use the dashboard for this we have made the user interface of admin dashboard very interactive.

Since it is a basic project so there are not much features that may make it more dynamic thanothers. But it may serve the basic purpose or making canteen digitalized. It is not suited for those who want easy to use databases with basic features.

Currently we have only password authentication in our project as a security feature, but infuture we are looking for adding more security features such as captcha, OTP.

In future we will be adding more services to our project and make it available to each andevery owner of cafeteria.

Currently we are providing only snacks but in future we will add food ordering module in it. Currently we do not provide Festive offers on snacks. But in future we are looking for addingFestive offers modules too.

References/ Bibliography

Websites

- www.stackoverflow.com
- www.youtube.com
- www.quora.com
- www.geeksforgeeks.org
- www.tutorialpoint.com
- www.javapoint.com