**INDUSTRY INTERNSHIP REPORT**

**ON**

**“ONLINE GROCERY STORE”**

**AT**

**ThinkNEXT Technologies Private Limited**

**S.C.F-113, Phase-11, Mohali**

**AN INDUSTRY INTERNSHIP REPORT SUBMITTED**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**

**FOR THE AWARD OF DEGREE OF**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**SUBMITTED BY:**

UJJWAL SHARMA

Roll Number:2020a1r116

**A picture containing text, clipart

Description automatically generated**

**SUBMITTED TO**

**COMPUTER SCIENCE AND**

**ENGINEERING**

**Model Institute of Engineering and Technology (Autonomous)**

**Jammu, India**

**2022**

**CANDIDATES’ DECLARATION**

I, **Ujjwal Sharma, 2020a1r116,** hereby declare that the work which is being presented in the Industry Internship Report entitled, “**Online Grocery Store**” in partial fulfillment of requirement for the award of degree of B.E. (Branch Name) and submitted in the Department Name, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me at “ThinkNEXT Technologies Private Limited S.C.F-113, Phase-11, Mohali” under the supervision and mentorship of **Ms. Ramandeep Kaur** Branch Head, ThinkNEXT Technologies Private Limited S.C.F-113, Phase-11, Mohali and Faculty **Mr. Sunil Kumar**(Trainer cum Developer, Software Development) respectively. The matter presented in this report has not been submitted to this or any other University / Institute for the award of B.E. Degree.

*Signature of the Student*  *Dated*:

**(Ujjwal Sharma) 3rd of October 2022**

**2020a1r116**

**INTERNSHIP CERTIFICATE**

****

**SUPERVISOR EVALUATION OF INTERN**

**Table

Description automatically generated**

**Table

Description automatically generated**

**COMPUTER SCIENCE AND ENGINEERING**

**Model Institute of Engineering and Technology (Autonomous) Kot Bhalwal, Jammu, India**

***(NAAC “A” Grade Accredited)***

**Ref. No.: 2020A1R116 Date: 03RD OCT 2022**

**CERTIFICATE**

Certified that this Industry Internship Report entitled **“ONLINE GROCERY STORE”** is the bonafide work of “**Ujjwal Sharma, 2020a1r116, of 5th Semester, Computer Science and Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu”,** who carried out the Industry Internship at “ThinkNEXT Technologies Private Limited S.C.F-113, Phase-11, Mohali” work under my mentorship during 20th July, 2022 to 29th August, 2022

**(Dr. Swati Goel)**

**Mentor-Internal Supervisor**

**Assistant Professor**

**Computer Science and Engineering, MIET**

*This is to certify that the above statement is correct to the best of my knowledge.*

**ACKNOWLEDGEMENTS**

Industry Internship is an important aspect in the field of engineering, where contributions are made by many people and organizations. The present shape of this work has come forth after contribution from different spheres.

Firstly, I would like to express my special thanks of gratitude to my teacher (Mr. Sunil Kumar) who gave me the golden opportunity to do this wonderful project on the topic (Python Django), which also helped me in doing a lot of Research and I came to know about so many new things I am thankful to them. I would also like to thank my parents, friends etc. who helped me in my Industry Internship.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

I must record my deep sense of gratitude to Prof. (Dr.) Ankur Gupta (Director, MIET) and Prof. (Dr.) Ashok Kumar (Dean Academics & HOD CSE, MIET) for their guidance, constant inspiration and encouragement, and for their keen involvement throughout the course of present work.

I express my sincere gratitude to “ThinkNEXT Technologies Private Limited S.C.F-113, Phase-11, Mohali” for giving me the opportunity to work on an Industry Internship during my 3rd year of B.E.

I express my sincere gratitude to Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity to work on Industry Internship during my 3rd year of B.E.

I express my sincere gratitude to ThinkNEXT Technologies Private Limited S.C.F-113, Phase-11, Mohali, Mohali and Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity. I perceive this opportunity as a big milestone in my career development.

I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

At the end thanks to the Almighty for everything

**(*Ujjwal Sharma*)**

**2020a1r116**

**SELF EVALUATION**

**Letter

Description automatically generated with medium confidenceTable

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

**ATTENDANCE REPORT**

**A picture containing table

Description automatically generated**

**ABSTRACT**

THIS PROJECT IS AIMED AT DEVELOPING A WEB APPLICATION THAT DEPICTS ONLINE SHOPPING OF GROCERY PRODUCTS. IT INTEGRATES THE BENEFITS OF ORDERING PRODUCTS WITH THE CONVENIENCE OF AN ONLINE EXCITEMENT AND GOING WITH TECHNOLOGY, MINUS THE COMMUTING HAZARDS AND EXPENSES. IT WILL USHER IN THE IMMENSE FLEXIBILITY AND SOPHISTICATION IN THE EXISTING MANUAL PLATFORM STRUCTURES, WITH THE PERFECT BLEND OF SYNCHRONOUS AND SYNCHRONOUS INTERACTION. IT PROVIDES A MEANS OF COLLABORATIVE E-ORDERING FOR THE USTOMERS. THE “ONLINE GROCERY STORE PROJECT” HAS BEEN DEVELOPED TO OVERRIDE THE PROBLEM PREVAILING IN THE PRACTICING MANUAL SYSTEM. THIS SOFTWARE IS SUPPORTED TO ELIMINATE AND, IN SOME CASES, REDUCE THE HARDSHIPS FACED BY THIS EXISTING SYSTEM. MOREOVER, THIS SYSTEM IS DESIGNED FOR THE PARTICULAR NEED OF THE COMPANY TO CARRY OUT OPERATION IN A SMOOTH AND EFFECTIVE MANNER.

**Contents**

Candidates’ Declaration i

Internship Certificate ii

Supervisor Evaluation of Intern iii

Certificate iv

Acknowledgement v

Self-Evaluation vi

Attendance Report vii

Abstract viii

Contents ix

List of Figures x

**Chapter 1 INTRODUCTION 1-3**

1.1Introduction to the system 1

1.2 Problem Definition 2

1.3 Objective 2

1.4 Goal 3

1.5 Need of system 3

**Chapter 2 Hardware and Software requirement 4-5**

2.1 Introduction 4

2.2 System Environment  4

2.3 Software Requirement 5

2.4 Hardware Requirements 5

**Chapter 3 System Analysis 6-7**

3.1 Purpose 6

3.2Project Scope 7

3.3 System Description 7

**Chapter 4 Implementation 8-11**

4.1Python 8

4.2 HTML 8

4.3Cascading Style Sheet (CSS) 9 4.4 JavaScript 10

4.5 Django 11

**Chapter 5 System design 12-18**

5.1 Use case diagram 12

5.2 Sequence Diagram 14

5.3 Data flow Diagram 15

5.4 ER diagram 17

**Chapter 6 Output Screen 19-25**

**Chapter 7 Coding 25-39**

**Chapter 8 Conclusion 40-41**

8.1 Advantages & Limitation 40

8.2 Future Scope 41

8.3 Conclusion 41

**REFERENCES 42**

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| **Figure No.** | **Caption** | **Page No.** |
| 1 | Use Case Diagram between ADMIN and SYSTEM | 12 |
| 2 | Use Case Diagram between USER and SYSTEM | 13 |
| 3 | Sequence Diagram for Administrator | 14 |
| 4 | Sequence Diagram for User | 15 |
| 5 | Data Flow Diagram of the Shopping Site | 16 |

**Chapter 1**

**Introduction to the System:**

The “Online Grocery Shop Project” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the need of the company to carry out operations in a smooth and effective manner.

This application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Online Grocery Shop Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user in concentrating on their other activities rather than concentrate on record keeping. Thus, it will help the organization in better utilization of resources.

Every food organization, whether big or small, has challenges to overcome and managing the information of groceries, customers, groceries stock, sales.

**Problem Definition:**

Managing your online grocery shopping system may seem tricky, but this is part of Customer service system (application support direct contact with customer)

To Manage the Online Grocery Product Ordering. It Helps Customers to Book products from anywhere. Also make payment on Delivery. It helps to People to Book desired products at their prefer time.

**Objective:**

This software helps customers to find different products according to category, price, and subcategories. It is designed in such a way that one can view all the updates of the products from any place online. The software will help in easily maintaining and updating products in the website for the administrator. Also, quick and easy comparison of different products for the customers.

**Goal:**

The project is basically targeted at those people who would like online shopping and to have Internet access.

Finally, buyers are curious about comparing the prices for various products according to our budget.

To make a database that is consistent, reliable and secure.

To provide correct, complete, ongoing information.

To develop a well-organized information storage system.

To make good documentation to facilitate possible future enhancements

**Need for the System:**

There is always a need for a system that will perform to purchasing grocery products online according to customer requirement.

This system will reduce the manual operation required to maintain all the records of booking information. And generates various reports for analysis. Main concept of the project is to enter transaction reports and to maintain customer records. Hence this software can be used in any grocery shop to maintain their records easily.

**Chapter 2**

**Hardware and Software Requirements:**

**Introduction:**

In this chapter we mentioned the software and hardware requirements, which are necessary for successfully running this system. The major element in building systems is selecting compatible hardware and software. The system analyst must determine what software package is best for the **“Online Grocery Store System”** and, where software is not an issue, the kind of hardware and peripherals needed for the final conversion.

**System Environment:**

After analysis, some resources are required to convert the abstract system into the real one. All the resources, which accomplish a robust

The hardware and software selection begins with requirement analysis, followed by a request for proposal and vendor evaluation.

Software and real system are identified. According to the provided functional specification all the technologies and its capacities are identified. Basic functions and procedures and methodologies are prepared to implement. Some of the Basic requirements such as hardware and software are described as follows: -

**Hardware and Software Specification**

**Software Requirements:**

* Technology: Python Django
* IDE: Visual Studio Code
* Client-Side Technologies: HTML, CSS, JavaScript, Bootstrap
* Server-Side Technologies: Python
* Data Base Server: SQLite
* Operating System: Microsoft Windows/Linux

**Hardware Requirements:**

* Processor: Pentium-III (or) Higher
* Ram: 64MB (or) Higher
* Hard disk: 80GB (or) Higher

**Chapter 3**

**System Analysis:**

### Purpose:

To manage the online shopping of grocery products. It helps customers to search and buy medicines from anywhere. Also make payment on delivery for it. It helps people to book desired products at their preferred time.

The online grocery shop system is available in the market that can serve customers to book/purchase grocery products online

**Project Scope:**

The project has a wide scope, as it is not intended for a particular organization. This project is going to develop generic software, which can be applied by any business organization. Moreover, it provides facilities to its customer. Also, the software is going to provide a huge amount of summary data. This web application involves almost all the features of Online Shopping. The Future implementation will be online help for the customers and chatting with website administrator.

**System Overview:**

The key features required in the system are as follows:

* **Login:** This module has a drop-down list box from where we have to select

**ADMIN or USER**. ADMIN has all the rights in the software including updating the status of his site. The other fields in login **are** username and password. If the username and password are correct, then it is directed to the next page.

* **New user:** This module is for the users who do not have an account. Here users are allowed to create an account to login. The account creation is done by filling in the registration form with user details such as name, phone, email etc.
* **Product:** This module has information regarding the medicines such as its name, category, subcategory, image, price information, its features etc. The **ADMIN** has the authority to Add, Delete, Update etc. The **USER** can only view the products available in the stock etc.
* **Search:** This module helps the customer to ease his search based on his budget or interest. The search can be done on different categories and subcategories like category, subcategory, name, price etc.

**Chapter 4**

**Implementation**

**Python**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

**HTML**

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end

**CASCADING STYLE SHEET (CSS)**

Cascading Style Sheets (CSS) are a collection of rules we use to define and modify web pages. CSS is like styles in Word. CSS allows Web designers to have much more control over their pages’ look and layout. For instance, you could create a style that defines the body text to be Verdana, 10 points. Later, you may easily change the body text to Times New Roman, 12 points by just changing the rule in the CSS. Instead of having to change the font on each page of your website, all you need to do is redefine the style on the style sheet, and it will instantly change on all the pages that the style sheet has been applied to. With HTML styles, the font change would be applied to each instance of that font and must be changed in each spot.

CSS can control the placement of text and objects on your pages as well as the look of those objects.

HTML information creates the objects (or gives objects meaning), but styles describe how the objects should appear. The HTML gives your page structure, while the CSS creates the “presentation”. An external CSS is just a text file with a .css extension. These files can be created with Dreamweaver, a CSS editor, or even Notepad.

The best practice is to design your web page on paper first so you know where you will want to use styles on your page. Then you can create the styles and apply them to your page.

**JavaScript**

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more like C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without COMMUNICATING with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

**Django**

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

**Chapter 5**

**System Design**

**Use Case Diagram:**

Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:

* The main purpose is to show the interaction between the use cases and the actor.
* To represent the system requirement from user’s perspective.
* The use cases are the functions that are to be performed in the module.

**ADMIN SYSTEM**

**Fig 1: Use Case Diagram between ADMIN and SYSTEM**

**Feedback**

**USER SYSTEM**

**Fig 2: Use Case Diagram between USER and SYSTEM**

**Login**

**Application**

**Database**

**Login**

**:Request**

**:Validate()**

**:executeQuery()**

**Response**

**Show Result**

**Failed:show()**

**Fig 3: Sequence Diagram for Administrator**

Diagram, schematic

Description automatically generated

**Fig 4: Sequence Diagram for User**

**Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an Information System. A data flow diagram can also be used for the visualization of Data Processing. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

A DFD represents the flow of data through a system. Data flow diagrams are commonly used during problem analysis. It views a system as a function that transforms the input into desired output. A DFD shows movement of data through the different transformations or processes in the system.

Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately influences the structure of the whole system from order to dispatch to restock how any system is developed can be determined through a dataflow diagram. The appropriate register is saved in the database and maintained by appropriate authorities.

Diagram

Description automatically generated

**Fig 5: Data Flow Diagram of the Shopping Site**

**Entity Relationship Diagrams (ER-Diagrams):**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes

An **entity-relationship model** (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

**Symbols used in this E-R Diagram:**

**Entity**: Entity is a “thing” in the real world with an independent existence. An entity may be an object with a physical existence such as person, car or employee. Entity symbol is as follows

**Attribute:** Attribute is a particular property that describes the entity. Attribute symbols are

**Relationship:** Relationship will be several implicit relationships among various entity types whenever an attribute of one entity refers to another entity type some relationship exits. Relationship symbol is:

**Key attributes:** An entity type usually has an attribute whose values are distinct for each individual entity in the collection. Such an attribute is called key attribute. Key attribute symbol is as follows

**Product**

**Member**

Request

**Order Detail**

Place Order

**Login**

Check the Login ID

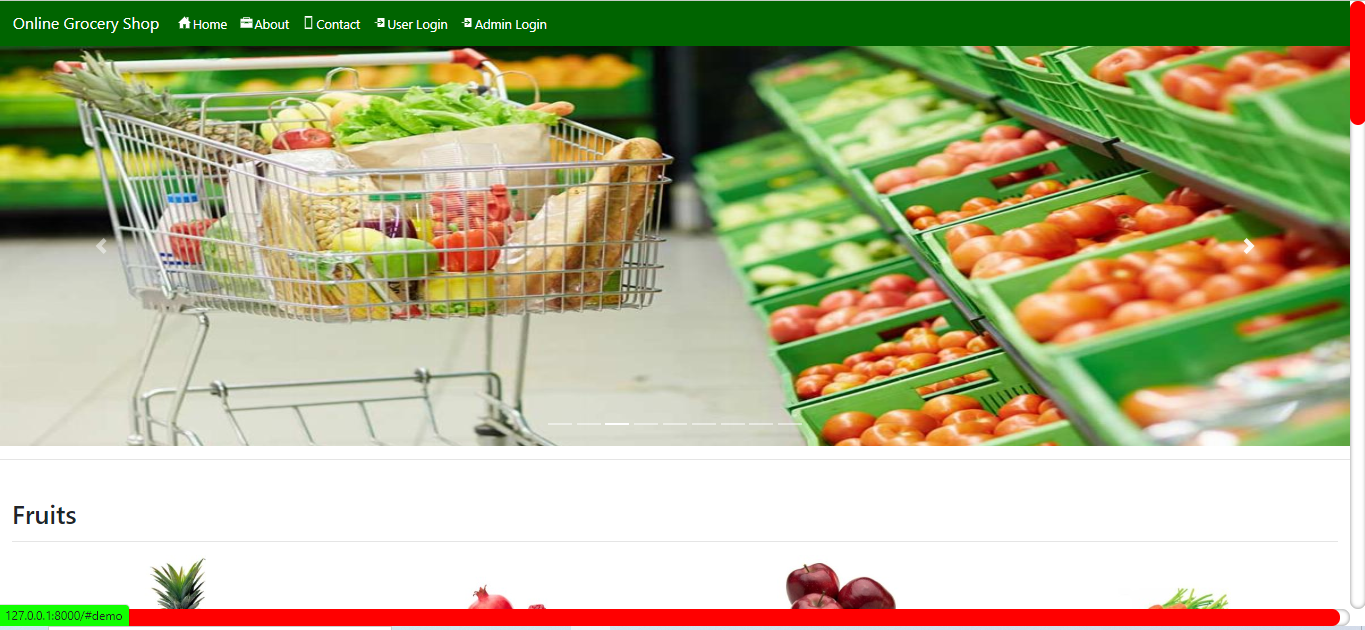
**Registration**

Req. New Member

**Chapter 6**

**Output Screen**

**HOME PAGE**

****

**USER LOGIN PAGE**

**Graphical user interface, application

Description automatically generated**

**PRODUCT CATEGORIES PAGE**

**Graphical user interface, application, Teams

Description automatically generated**

**VIEW CART PAGE**

**Graphical user interface, text, website

Description automatically generated**

**VIEW PRODUCTS PAGE**

**Graphical user interface, application

Description automatically generated**

**CHANGE PASSWORD PAGE**

**Graphical user interface, text, application

Description automatically generated**

**VIEW PROFILE PAGE**

**Graphical user interface, application, website

Description automatically generated**

**SEND FEEDBACK PAGE**

**Graphical user interface, application

Description automatically generated**

**ADMIN HOME PAGE**

**Graphical user interface, text, application

Description automatically generated**

**VIEW PRODUCT PAGE**

**Graphical user interface

Description automatically generated**

**VIEW CATEGORY PAGE**

**Timeline

Description automatically generated**

**VIEW BOOKING PAGE**

**A picture containing graphical user interface

Description automatically generated**

**VIEW USERS PAGE**

**Table

Description automatically generated with medium confidence**

**Chapter 7**

**Coding**

**USER LOGIN PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

{% load static %}

{% if error == 'yes' %}

<script>

    alert('logged in successfully');

    window.location.href = "/";

</script>

{% endif %}

{% if error == 'not' %}

<script>

    alert('Username & Password are not Matching');

</script>

{% endif %}

<div class="container">

    <h2 style="margin-top:20px" *align*="center">User Login Form</h2><hr>

<div class="container-fluid" style="width:70%;margin-top:10%">

<form method="post" action="">

    {% csrf\_token %}

  <div class="form-group">

    <label for="exampleInputEmail1">Username</label>

    <input type="text" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name="uname" placeholder="Enter Username">

  </div>

  <div class="form-group">

    <label for="exampleInputPassword1">Password</label>

    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password" name="pwd">

  </div>

  <button type="submit" class="btn btn-primary">Submit</button>

  <button class="btn btn-primary"><a href="{% url 'signup' %}" style="color:white">Signup</a></button>

</form>

</div>

</div>

{% endblock %}

**SIGNUP PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

{% load static %}

<div class="container">

    <h2 style="margin-top:20px" *align*="center">Registration Form</h2><hr>

<div class="container-fluid" style="width:70%;margin-top:10%">

<form method="post" action="" enctype="multipart/form-data">

    {% csrf\_token %}

    <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputEmail4">First Name</label>

      <input type="text" class="form-control" placeholder="Enter First Name" name="fname" id="inputEmail4">

    </div>

    <div class="form-group col-md-6">

      <label for="inputPassword4">Last Name</label>

      <input type="text" class="form-control" name="lname" placeholder="Enter Last Name" id="inputPassword4">

    </div>

  </div>

    <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputEmail4">User Name</label>

      <input type="text" class="form-control" name="uname" placeholder="Enter User Name" id="inputEmail4">

    </div>

    <div class="form-group col-md-6">

      <label for="inputPassword4">Password</label>

      <input type="password" class="form-control" name="pwd" id="inputPassword4">

    </div>

  </div>

        <div class="form-row">

<div class="form-group col-md-6">

    <label for="exampleInputEmail1">Email</label>

    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name="email" placeholder="Enter Email">

  </div>

  <div class="form-group col-md-6">

    <label for="exampleInputPassword1">Contact</label>

    <input type="text" class="form-control" id="exampleInputPassword1" placeholder="Enter Contact Detail" name="contact">

  </div>

        </div>

        <div class="form-row">

    <div class="form-group col-md-6">

    <label for="exampleInputPassword1">City</label>

    <input type="text" class="form-control" id="exampleInputPassword1" placeholder="Enter City" name="city">

  </div>

    <div class="form-group col-md-6">

        <label for="exampleInputPassword1">Date of Birth</label>

    <input type="date" class="form-control" id="exampleInputPassword1" name="date">

  </div>

        </div>

    <div class="form-row">

    <div class="form-group col-md-6">

    <label for="exampleInputPassword1">Full Address</label>

    <input type="text" class="form-control" id="exampleInputPassword1" placeholder="Enter Address Detail" name="add">

  </div>

<div class="form-group col-md-6">

    <label for="exampleInputPassword1">Image</label>

    <input type="file" class="form-control" id="exampleInputPassword1" name="img" value="choose file">

</div></div>

  <button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</div>

{% if error %}

<script>

    alert('Regitered Successfully');

    window.location = "{% url 'login' %}";

</script>

{% endif %}

{% endblock %}

**VIEW CART PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

<*center*>

    <h2 style="color:red;margin-top:2%">View Cart</h2>

</*center*>

<hr>

{% if not cart %}

<h1 *align*="center" style="color:blue">{{message}}</h1>

{% else %}

{% for i in cart %}

<*center*>

    <style>

        .container {

            border: 1px solid black;

            border-radius: 6px;

            margin-bottom: 5%;

        }

        .container1,

        .contain {

            border: 1px solid black;

            border-radius: 6px;

            width: 70%;

            border: 1px solid black;

        }

    </style>

    <div class="container" style="margin-top:2%;height:150px;width:80%">

        <div class="row" style="height:90%">

            <div class="col-md-4" style="height:120px">

                <img src="{{i.product.image.url}}"

                    style="width:30%;height:100px;border:1px solid darkgray;margin-top:5%">

            </div>

            <div class="col-md-4" style="height:100px">

                <h5 style="margin-top:5%">{{i.product.name}}</h5>

                <hr>

                <h6>Price : Rs.{{i.product.price}}</h6>

                <p>{{i.product.desc}}</p>

            </div>

            <div class="col-md-4" style="height:100px;">

                <*center*><a href="{% url 'remove\_cart' i.id %}" style="margin-top:15%"

                        class="btn btn-sm btn-danger text-light text-capitalize">

                        <ion-icon name="trash"></ion-icon>Remove

                    </a></*center*>

            </div>

        </div>

    </div>

</*center*>

{% endfor %}

|<div class="container" style="border:1px solid white">

    <div class="row">

        <div class="col-md-4" style="height:100px;">

            <*center*><a href="{% url 'view\_product' 0 %}" style="margin-top:1%;width:100%"

                    class="btn btn-lg btn-info text-light text-capitalize">

                    <ion-icon style="margin-right:8px" name="pricetags"></ion-icon>Continue Shopping

                </a></*center*>

        </div>

        <div class="col-md-4" style="height:100px;">

            <*center*><a href="{% url 'booking' book %}" style="margin-top:1%;width:50%"

                    class="btn btn-lg btn-info text-light text-capitalize">

                    <ion-icon style="margin-right:8px" name="cash"></ion-icon>Buy Now

                </a></*center*>

        </div>

        <div class="col-md-4" style="height:100px;">

            <*center*><a style="margin-top:1%;width:100%;border:1px solid black"

                    class="btn btn-default btn-lg text-dark text-capitalize">Total : {{total}}</a></*center*>

        </div>

    </div>

    {% endif %}

</div>

{% endblock %}

**ADMIN HOME PAGE CODING**

{% extends 'navigation.html' %}

{% load static %}

{% block body %}

<style>

  .glow {

    font-size: 80px;

    color: #fff;

    text-align: center;

    -webkit-animation: glow 1s ease-in-out infinite alternate;

    -moz-animation: glow 1s ease-in-out infinite alternate;

    animation: glow 1s ease-in-out infinite alternate;

  }

  @keyframes glow {

    from {

      text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #e60073, 0 0 40px #e60073, 0 0 50px #e60073, 0 0 60px #e60073, 0 0 70px #e60073;

    }

    to {

      text-shadow: 0 0 20px #fff, 0 0 30px #ff4da6, 0 0 40px #ff4da6, 0 0 50px #ff4da6, 0 0 60px #ff4da6, 0 0 70px #ff4da6, 0 0 80px #ff4da6;

    }

  }

  /\* Rounded border \*/

  hr.rounded {

    border-top: 8px solid lightblue;

    border-radius: 5px;

  }

</style>

</style>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

  integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<br>

<*center*>

  <div class="container" style="margin-top:10%;height:150px;width:80%">

    <div class="row" style="height:90%">

      <div class="col-md-4">

        <a href="{% url 'view\_user' %}">

          <h1>Total Customer</h1>

          <h2 style="color:blue">{{total\_customer}}</h2>

          <hr class="rounded">

        </a>

      </div>

      <div class="col-md-4">

        <a href="{% url 'admin\_viewBooking' %}">

          <h1>Total Booking</h1>

          <h2 style="color:blue">{{total\_book}}</h2>

          <hr class="rounded">

        </a>

      </div>

      <div class="col-md-4">

        <a href="{% url 'admin\_view\_product' %}">

          <h1>Total Product</h1>

          <h2 style="color:blue">{{total\_pro}}</h2>

          <hr class="rounded">

        </a>

      </div>

    </div>

  </div>

</*center*>

<footer class="site-footer">

  <div class="container">

    <div class="row pt-5 mt-5 text-center">

      <div class="col-md-12">

        <p style="font-size : 18pt ; color : #FF0000 ; font-weight : bold">

          Copyright &copy;

          <script>document.write(new Date().getFullYear());</script> All Rights Reserved <span

            style="color : brown">|</span> <span style="color : #006400">This website is made with <i

              class="fas fa-heart icon-heart text-warning" aria-hidden="true"></i> by </span><a

            style="color:rgb(1, 4, 7) ;" href="#" target="\_blank">Ujjwal Sharma</a>

        </p>

      </div>

    </div>

  </div>

</footer>

{% endblock %}

**VIEW BOOKING PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

<*center*><h2 style="color:red;margin-top:2%">View Booking Detail</h2></*center*><hr>

{% for i in product %}

{% if i.id in book %}

<*center*>

    <style>

        .container{

            border:1px solid black;

            border-radius:6px;

            margin-bottom:5%;

        }.container1 ,.contain{

            border:1px solid black;

            border-radius:6px;

            width:70%;

            border:1px solid black;

        }

    </style>

<div class="container" style="margin-top:2%;height:150px;width:80%">

    <div class="row" style="height:90%">

        <div class="col-md-4" style="height:120px">

            <img src="{{i.image.url}}" style="width:30%;height:100px;border:1px solid darkgray;margin-top:5%">

        </div>

        <div class="col-md-4" style="height:100px">

            <h5 style="margin-top:5%">{{i.name}}</h5><hr>

        <h6>Price : Rs.{{i.price}}</h6>

        <p>{{i.desc}}</p>

        </div>

            <div class="col-md-4" style="height:100px;">

                <h4 style="margin-top:15%;color:blue">Booked Successfully</h4>

        </div>

    </div>

</div>

    </*center*>

{% endif %}

{% endfor %}

<div class="container" style="border:1px solid white">

<div class="row">

    <div class="col-md-6" style="height:100px;">

                <*center*><a style="margin-top:1%;width:100%;border:1px solid black" class="btn btn-default btn-lg text-dark text-capitalize">Total : {{total.total}}</a></*center*>

            </div>

</div>

<*center*><h2 style="color:red;margin-top:2%">Customer Detail</h2></*center*><hr>

<div class="row">

    <div class="col-md-6" style="height:100px;"><hr>

        <h4>Name :  {{profile.user.first\_name}} {{profile.user.last\_name}}</h4><hr>

        <h4>Email :  {{profile.user.email}}</h4><hr>

        <h4>Address : {{profile.address}}</h4><hr>

        <h4>Contact : {{profile.contact}}</h4><hr>

    </div>

    <div class="col-md-6" style="height:100px;"><hr>

        <h4>  <img src="{{profile.image.url}}" style="width:150px;height:140px"></h4>

    </div>

</div>

</div>

{% endblock %}

**VIEW PRODUCTS PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

<*center*>

  <h2 style="color:red;margin-top:2%">View Product</h2>

</*center*>

<hr>

<div class="container" style="margin-top:4%">

  <table class="table table-bordered" id="myTable">

    <thead>

      <tr>

        <th>Product Name</th>

        <th>Image</th>

        <th> Category</th>

        <th>Price</th>

        <th> Description</th>

        <th>Action</th>

      </tr>

    </thead>

    <tbody>

      {% for i in pro %}

      <tr>

        <td>{{i.name}}</td>

        <td><img src="{{i.image.url}}" style="width:50px;height:50px"></td>

        <td>{{i.category.name}}</td>

        <td>{{i.price}}</td>

        <td>{{i.desc}}</td>

        <td><a href="{% url 'delete\_product' i.id %}" onclick="return confirm('Are you sure?')"><button

              class="button button1">Delete</button></a></td>

      </tr>

      {% endfor %}

    </tbody>

    {{count}}

  </table>

</div>

<script>

  {% for i in message %}

  alert("{{i}}")

  {% endfor %}

</script>

{% endblock %}

**ADD PRODUCT PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

{% load static %}

<div class="container">

    <h2 style="margin-top:20px" *align*="center">Add Product</h2><hr>

<div class="container-fluid" style="width:70%;margin-top:10%">

<form method="post" action="" enctype="multipart/form-data">

    {% csrf\_token %}

    <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputEmail4">Product Name</label>

      <input type="text" class="form-control" placeholder="Enter First Name" name="pname" id="inputEmail4">

    </div>

    <div class="form-group col-md-6">

    <label for="exampleInputPassword1">Product Image</label>

    <input type="file" class="form-control" id="exampleInputPassword1" name="img" value="choose file">

</div></div>

    <div class="form-row">

        <div class="form-group col-md-6">

      <label for="inputState">Category</label>

      <select id="inputState" class="form-control" name="cat">

          {% for i in cat %}

        <option value="{{i.name}}">{{i.name}}</option>

          {% endfor %}

      </select>

    </div>

    <div class="form-group col-md-6">

      <label for="inputEmail4">Product Price</label>

      <input type="number" class="form-control" name="price" placeholder="Enter User Name" id="inputEmail4">

    </div></div>

    <div class="form-group">

    <label for="exampleFormControlTextarea1">Description</label>

    <textarea class="form-control" id="exampleFormControlTextarea1" name="desc" rows="3"></textarea>

  </div>

  <button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</div>

{% if error %}

<script>

    alert('1 Product Added Successfully');

</script>

{% endif %}

{% endblock %}

**ADD CATEGORY PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

{% load static %}

{% if error %}

<script>

    alert('One New Category Added Successfully');

</script>

{% endif %}

<div class="container">

    <h2 style="margin-top:20px" *align*="center">Add Category</h2><hr>

<div class="container-fluid" style="width:70%;margin-top:10%">

<form method="post" action="">

    {% csrf\_token %}

  <div class="form-group">

    <label for="exampleInputEmail1">Category Name</label>

    <input type="text" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name="cat" placeholder="Enter Category">

  </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</div>

{% endblock %}

**USER EDIT PROFILE PAGE CODING**

{% extends 'navigation.html' %}

{% block body %}

{% load static %}

<div class="container">

    <h2 style="margin-top:20px" *align*="center">Edit Profile</h2><hr>

<div class="container-fluid" style="width:70%;margin-top:10%">

<form method="post" action="" enctype="multipart/form-data">

    {% csrf\_token %}

    <div class="form-row">

    <div class="form-group col-md-6">

      <label for="inputEmail4">First Name</label>

      <input type="text" class="form-control" value="{{pro.user.first\_name}}" name="fname" id="inputEmail4">

    </div>

    <div class="form-group col-md-6">

      <label for="inputPassword4">Last Name</label>

      <input type="text" class="form-control" name="lname" value="{{pro.user.last\_name}}" id="inputPassword4">

    </div>

  </div>

    <div class="form-row">

    <div class="form-group col-md-12">

      <label for="inputEmail4">User Name</label>

      <input type="text" class="form-control" name="uname" value="{{pro.user.username}}" id="inputEmail4" readonly>

    </div>

  </div>

        <div class="form-row">

<div class="form-group col-md-6">

    <label for="exampleInputEmail1">Email</label>

    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" name="email" value="{{pro.user.email}}">

  </div>

  <div class="form-group col-md-6">

    <label for="exampleInputPassword1">Contact</label>

    <input type="text" class="form-control" id="exampleInputPassword1" value="{{pro.contact}}" name="contact">

  </div>

        </div>

        <div class="form-row">

    <div class="form-group col-md-12">

    <label for="exampleInputPassword1">City</label>

    <input type="text" class="form-control" id="exampleInputPassword1" value="{{pro.city}}" name="city">

  </div>

        </div>

    <div class="form-row">

    <div class="form-group col-md-6">

        <label for="exampleInputPassword1">Date of Birth</label>

    <input type="date" class="form-control" id="exampleInputPassword1" name="date">

  </div>

        <div class="form-group col-md-6">

        <label for="exampleInputPassword1">Date of Birth</label>

    <input type="text" class="form-control" id="exampleInputPassword1" value="{{pro.dob}}" readonly>

        </div>

        </div>

    <div class="form-row">

    <div class="form-group col-md-12">

    <label for="exampleInputPassword1">Full Address</label>

    <input type="text" class="form-control" id="exampleInputPassword1" value="{{pro.address}}" name="add">

  </div>

    </div>

    <div class="form-row">

<div class="form-group col-md-6">

    <label for="exampleInputPassword1">Image</label>

    <input type="file" class="form-control" id="exampleInputPassword1" name="img" value="choose file">

</div>

        <div class="form-group col-md-6">

    <img src="{{pro.image.url}}" style="width:150px;height:150px">

</div>

    </div>

  <button type="submit" class="btn btn-primary" style="margin-bottom:5%">update</button>

</form>

</div>

</div>

{% if error %}

<script>

    alert('Update Successfully');

    window.location = "{% url 'profile' %}";

</script>

{% endif %}

{% endblock %}

**Chapter 8**

**Conclusion**

**Advantages of “Online Grocery Store System”:**

“Online Grocery Store System” provides various features, which complement the information system and increase the productivity of the system. These features make the system easily usable and convenient. Some of the important features included are listed as follows:

* Intelligent User Forms Design
* Data access and manipulation through same forms
* Access to most required information
* Data Security
* Restrictive data access, as per login assigned only.
* Organized and structured storage of facts.
* Strategic Planning made easy.
* No decay of old Records.
* Exact financial position of the Business.

**Limitations of “Online Grocery Store System”:**

Besides the above achievements and the successful completion of the project, we still feel the project has some limitations, listed below:

* 1. It is not a large-scale system.
  2. Only limited information is provided by this system.
  3. Since it is an online project, customers need internet connection to buy products.

People who are not familiar with computers can’t use this software.

**FUTURE SCOPE**

Thisweb application involves almost all the features of online shopping. The future implementation will be online help for the customers and chatting with the website administrator.

Also, since the deliveries from these local vendors will not be as time-consuming as these days Flipkart, Amazon, etc. take but rather will be delivered the same day of an order placed. Else the shopkeeper can ask the customer if the product will be available by the next day, so if he/she still wants to place the order, it can be done.

Again, return or exchange will be easy since the delivery boy can even do it as the store is nearby. Including a chat box for public benefit is also a great idea via which people can directly have a conversation with some officials regarding any type of queries

**CONCLUSION**

The project entitled “Online Grocery Store” is developed using HTML, CSS and Bootstrap as front end and Python Django and SQLite database in back end to computerize the process of online buying of grocery products. This project covers only the basic features required.

**References**

<https://docs.python.org/3/tutorial/>

<https://www.pythontutorial.net/>

<https://www.geeksforgeeks.org/python-programming-language/learn-python-tutorial/>

<https://www.w3schools.com/python/>