**CHAPTER - I**

**INTRODUCTION**

The food industry is highly labor intensive and the biggest expense in the food industry is the cost of employing the right kind of people to do the work. Labor rates are steadily rising year in and year out, making it difficult to hire labor. One of the ways to reduce this expense is to use modern technology to replace some of the jobs done by human beings and make machines do the work. Here we propose an “Online Food Ordering Management System” that has been designed for Fast Food restaurants, Take-Out or College Cafeterias. The system may well be implemented in any organization that distributes foodstuff. Because the whole process of accepting orders is automated, the meal ordering experience for both the client and the restaurant is optimized. Online food real time ordering management system's objective is to give customers a way to order food and drinks over the internet.

**Overview of Project**

Ordering is a process of the customers specifying what they want, so that the order can be recorded by using a note, form, computer system and many others, followed by passing it to the relevant department for processing and finally delivery of the services or  products to the customers based on the order. An ordering system is referred as a set of detail methods that is being used in handling the ordering process Food ordering can be computerized or done manually. A computerized ordering system or more often known as Ordering Management System (OMS) can be defined in several ways. An ordering management system is a computer software system used in a number of industries for order entry and processing. To discover information about this project, it is planned to having visits to the restaurant for observing the current ordering system. Besides that, browsing restaurant official websites is a must. Browsing through those websites can help to gather some information, such as, about the restaurants promotions, available foods and the prices. A research on food ordering system is considered to get better understanding about the system. This can be done by reading through the journals and articles. Nevertheless, reviewing up-to-date documentation such as restaurants menu and receipts will also be included in the plan.

* 1. **Existing System**

In the present scenario people have to physically visit the hotels or restaurants for eating food, and have to make payment through cash mode most of the times due to unawareness of advanced technologies at certain places.

In this method time as well as physical work is required, among which time is something that no one has in ample amount. The traditional food ordering procedure is not efficient enough for hotels and restaurant, as they have to deal with crowd, in their restaurant.

The old methods can be classified into categories which are paper grounded and verbal grounded. For paper based work, the waiter comes and pens down foods that customers order and pass the food list containing paper to the chefs or cooks in the kitchen for further process.

Also from the owner’s point of view maintaining data record and the accounts in physical file is cumbersome and tedious work to do. And also it is full of risk as anyone can access it and modify the data.

* 1. **Drawbacks of Existing System**

While there are many advantages to the online food ordering system, there are also some disadvantages to online food ordering systems. They are

**Price:**

One of the major drawbacks of online food ordering systems is price. When food is ordered for more than one person, the cost is usually equal to eating at a good restaurant every night. Many food ordering systems cost more than Rs.200 per person per day. Even more expensive for some other food ordering systems. For individuals with a limited food budget, online food ordering systems are often too expensive.

**Limited menu:**

Another disadvantage for food ordering systems is menu choices. Most food ordering systems have a limited number of meals. The menu changes every few weeks or months, but if you stick to the system for more than a few months the menu items will come back again and again. You should also eat the food provided for that week. If you do not want to eat that particular food, you may have to order another food from another place or eat food you do not like.

**Preparation:**

The preparation factor may be a disadvantage to food ordering systems. Most food ordering systems give frozen food. They are usually easy to prepare, but they usually take more than an hour to cook because the food is frozen. To avoid long cooking times, you can remove the food from the freezer the day before. However, remember to eliminate food from the freezer to reduce cooking time.

**Quality of food may be suffered:**

One problem with the food ordering system is that the quality of the food served is often worse than eating at a restaurant. Often, food has to be feed over long distances, and over time, precious vitamins can be lost. Also, food from the ordering system is often served in plastic packaging, which may not be very appealing to your eyes compared to the food neatly placed on your plate in a restaurant.

**Food may get cold:**

Due to the long ordering distances, your food may also be cold once it is finally delivered to your home. You need to reheat it or eat it cold.

This is especially true, if you order in an emergency the streets are often crowded and the ordering person will be stuck in traffic.

**The vibe of the restaurant is missing:**

* In some restaurants, there is also a good circumstance which you will miss if you order your food at home.
* For example, if you spend your evening in a good Chinese restaurant, you will often feel like you are actually in China because the decoration and the whole atmosphere are in line with the Chinese way of life. If you order food at your home, you will lose all of these.
* Also, from time to time, it would be great if you could take your partner or family to a nice restaurant for dinner to spend a good evening.
  1. **Proposed System**

The simulation first starts with the admin entering his/her credentials (ID and password). Once that has been verified, the admin can access the main admin panel where he/she can edit the categories, the food items as well can view the orders placed and reservations made. Now we get a window that displays the order number, customer ID, food name, price and quantity. Once the customer finalizes his/her order, they are asked to enter their name, address and other contact details where the total price is displayed and the customer can click the 'order now' button to get a message of confirmation of order. Once you enter the admin portal, you get the option of adding food, deleting food or updating food. Any option of choice leads you to the food menu. Once the selected operation is carried out, the end result, i.e., the added food or the updated food list is displayed and if you have deleted a food, that particular food disappears from the main menu on the website which will be visible to the customer.

* 1. **Scope of the system**

This represents the set of customers, which are the clients who will be using this application. The customers are for whom the system is being designed.

**Its attribute set includes:**

* **Name:** This is the name of the customer, searching or purchasing the products. When signing up to the website the name of customer is stored, this is done for the future referencing and maintaining the user’s data record (history). It is the composite attribute which contains two more attributes that is First\_Name and Last\_Name. That contains user’s first name and last name.
* **Cus\_id:** This is the identity number assigned by the admin to the users so as to identify them uniquely in future. This identity number is helpful in fetching data of individual user from a big set. This is mainly to manage the huge database system where the entire data is being stored. It is a permanent identity number given by the admin to the customer to maintain customer history.
* **Cus\_order\_id**: This is the identity number given to determine and manage the sequence of servicing. Since multiple customer will place orders, so as to schedule whom to give the delivery first is determined by the help of this number, so as to maintain consistency in the system working procedure. It will be unique for each order a day. But the same id can be repeated on a new day, as it is mainly for the restaurant’s reference and to prevent any type of conflict.

**1.5.1 Address**

This field is for the physical address of the customer where the restaurant authority is required to deliver the parcels. It may or may not the same as customer’s permanent address or resident, but can be the office place or any place.

**Its attribute includes:**

* **Address\_id:**An identity through which categorization of places may be done. As address may or may not be unique for each customer registered. But still this identity helps the delivery person to identify the right place to deliver.
* **Zip\_code:**  It is the pin code or the postal code of a region, and which is utmost important in any address, since multiple places, streets, bungalows with same name exist. This is even important in any national level identification of address. Also this will help the owner in surveying that which region has their more demand so as to expand their business in that region.
* **Phone:** The users contact number is something that must be correct, because if at some point of time delivery person gets confuse with the address, it can be used for confirmation. Also the restaurant authority can contact to their customers for any type of feedbacks or to know the delivery service is good or not.

## 1.5.2 Orders

The customers places order, which is not only till here, there are some work that need to be done in the database in order to maintain records for keeping track on monthly basis.

* **Order\_id:**This is the identity number given to determine and manage the sequence of servicing. Since multiple customer will place orders, so as to schedule whom to give the delivery first is determined by the help of this number, so as to maintain consistency in the system working procedure. It will be unique for each order a day. But the same id can be repeated on a new day, as it is mainly for the restaurant’s reference and to prevent any type of overlapping of thoughts between customers and owners. It is mainly for the chef’s reference.
* **Cus\_id:** This is the identity number assigned by the admin to the users so as to identify them uniquely in future. This identity number is helpful in fetching data of individual user from a big set. This is mainly to manage the huge database system where the entire data is being stored. It is a permanent identity number given by the admin to the customer to maintain customer history.
* **Total\_price:**This attribute manages the total price sum of the orders user has made in one attempt. It is one of the most important attribute, since most of the times people change their menu order list contents depending upon their needs, health and economic situation.
* **Timing:**Time is something most important to be valued. And one of the major reason behind the success of this food ordering system. So managing this cause becomes a goal to be completed. In order to maintain the business work better, the authority must stick to its commitment.

## 1.5.3 Payment

It defines the payment to be done by the customer for order placed from the web store at worth price. Also various security encryption mechanisms have been used, so the customer details of accounts and other credentials are safe and secure.

* **Payment\_type:**The user is provided with lots of options that he/she can opt for making the payment depending upon their ease. There are many choices available like net banking, use of wallets like paytm and i-cash cards, also the credit card and debit card options are available too.
* **Payment\_id:**It is for the benefit of user as well as the website owners, since the payment\_id is helpful in maintaining the payment record in the database, as well as it is also provided to the customer after the successful completion of payment. As later customer can claim anytime that they have already done the payments and the owners cannot deny. So it is useful to prevent any kind of fraud from both the sides.
* **Price:** It is the record of the total sum amount the user needs to pay, and after the payment, it is used to update in the server-side database to keep the record of the net profit or loss on daily basis.

## 1.5.4 Worker

The base of any company, restaurant or hotel is its employee. It is said that an Organization is known by its employee and work. Employees will work honestly and with complete dedication if they are paid sufficient enough money. On the whole it’s just like a food cycle, everyone depends on somebody.

* **Worker\_name:** The name of worker is important to maintain their database of work and payment record. Also if any complaints are filed then it is required.
* **Timing:**Time is something most important to be valued. And one of the major reason behind the success of this food ordering system. So managing this cause becomes a goal to be completed. In order to maintain the business work better, the authority must stick to its commitment. Workers are paid for their good work and more than that for completion of work before time.
* **Worker\_id:**To uniquely identify each worker and prevent any type of redundancy in records.
* **Salary:**The amount of money to be paid to the workers for their effective and on time work done.

## 1.5.5 Delivery mode

The delivery sequence and choice is not same for everyone but varies person to person. It may happen that even some time a person says no to home delivery as he/she is passing by and can pick the parcel themselves. But it is almost an ideal case.

* **Urgent:**In the some cases like un invited guest arrival, late night, people prefer to pay more and get the order delivered urgently. So restaurants manage such situations by not following the sequence of order place, as they are getting more than usual. And with other customer whom they have delayed they manage it with some small gifts or offers.
* **Normal:**The usual mode of delivery, that is followed by the sequence of orders placed. It is the normal and majority case. The hotels manage do not need to put an extra efforts to manage these.

**1.5.6 Reports**

Reports used for management and daily operations will be produced.

Ex:

• Pending payments

• Payments

Ability to filter reports based on certain criteria will also provide under this section. User will be having the option to print the reports.

**1.5.7 Application control**

Control over the application will be available under two sub categories.

**Menu access control**

Access level for each menu item can be controlled by the administrator without accessing the back end (source codes). Further ability to disable, enable or change the name of a menu item is enables by the system.

**User access control.**

User access is controlled by user name and password. Changing the password will be provided by the system. Further user level of each user can be changed by administrator.

In next few chapters more details are given on the modules of the proposed system, design phase and the present situation of the system

* 1. **Aims and objectives of the dissertation**

Dissertation is prepared a way that depicts the problem domain, requirements and how the system is developed to address the problem definition. Further all the techniques and methodologies used are mentioned and explained in further chapters accordingly.

**CHAPTER - II**

**BACKGROUND OF STUDY**

The online food ordering system is one of the latest servicers most fast food restaurants in the western world are adopting. With this method, food is ordered online and delivered to the customer. This is made possible through the use of electronic payment system. Customers pay with their credit cards, although credit card customers can be served even before they make payment either through cash or check. So, the system designed in this project will enable customers go online and place order for their food.

Due to the great increase in the awareness of internet and the technologies associated with it, several opportunities are coming up on the web. So many businesses and companies now venture into their business with ease because of the internet. One of such business that the internet introduced is an online food ordering system. In today’s age of fast food and take out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until recently, most of this delivery orders we replaced over the phone, but there are many disadvantages to this system.

**2.1 STATEMENT OF PROBLEM**

As industries are fast expanding, people are seeking for more ways to purchase products with much ease and still maintain cost effectiveness. The vendors need to purchase the products in order to sell to end users. The manual method of going to their local food sales outlets to purchase food is becoming obsolete and more tasking. Food can be ordered through the internet and payment made without going to the restaurant or the food vendor. So there is need for a wide range of publicity and enabling direct order, processing and delivering of food through online system. For this system, there will be a system administrator who will have the rights to enter the menu with current prevailing prices.

**2.2 OBJECTIVES OF STUDY**

This study lays out a framework for a new system to be developed and brought to the market for maximum use and to create an avenue through the web where users can log on to our server and make a selection of whatever goods or food they like and subsequently pay via the internet. The following are the objectives this would bring:

1. The home page of this web interfile provides an avenue where customers will be able to gather more and reliable information about what the fast food industry really does.
2. The products and services offered would provide the customers with all the different categories of available products that they can choose and select from.
3. This will provide a user friendly environment between the customer and employee thus increasing the efficiency of the food ordering system.
4. There will also be an online purchase form with which valued customers will be using to get in touch with any of their request whenever the need arises.
5. It will also help for easy retrieval of orders made by the customers.

**2.3 SCOPE OF STUDY**

In this project, a fast food company is designed and THE SPICE HOUSE is taken as a case study to enable customers order for food and get it delivered accordingly and also to reduce the long queues of customers at the counter ordering for food and to reduce the work lord on the employees. The following things are among other things that are discussed and what the software would handle:

1. About the fast food company
2. The fast food and the services offered there
3. Online purchase
4. Type of food provided.

**2.4 SIGNIFICANCE OF STUDY**

In view of the rapid development of computer technology in almost all the fields of operation and its use in relation to information management, it has become important to look into the development of online ordering system for firms to meet up with demands of the customers. Therefore, the food ordering and delivery system will help customers and management to:

1. Advertise available foods in their company.
2. Reduce the work load in the present system.
3. Reduce time wasted in data processing
4. Create a platform for online purchase and delivery of fast food.
5. Keep accurate record on purchased order and delivery.

**2.5 LIMITATIONS**

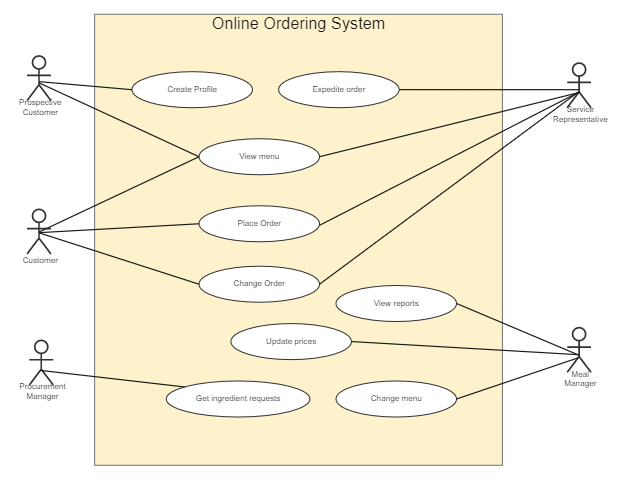
Due to time and financial constraints, the software that is developed covers only the aspect of food ordering and payments.

**CHAPTER - III**

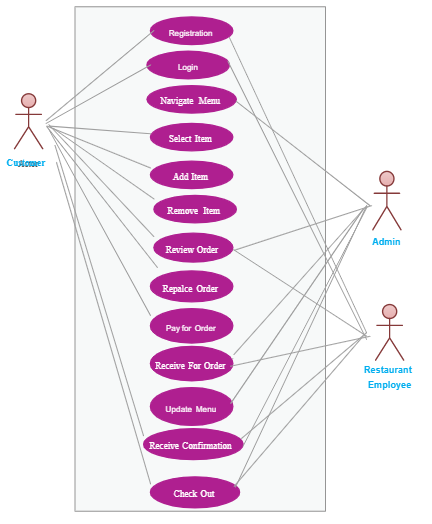
**ANALYSIS AND DESIGN/ METHODOLOGY**

Prior to analysis basic data gathering process was done and several data gathering techniques were used.

**3.1 USE CASE DIAGRAM**

****

**3.2 ER DIAGRAM**

****

**3.3 Iterative and incremental development**

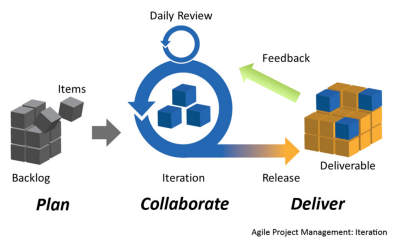
Iterative and Incremental development is any combination of both iterative design or iterative method and incremental build model for software development. The combination is of long standing and has been widely suggested for large development efforts.

**3.3.1 Overview**

The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. A teach iteration, design modifications are made and new functional capabilities are added.

The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straight forward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light weight iterative project the code may represent the major source of documentation of the system, however, in a critical iterative project a formal Software Design Document may be used. The analysis of iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.



**3.3.2 How the theories were applied**

As already mentioned above, the system has categorized in to 3 major modules as mentioned bellow.

1. Admin Login
2. User login
3. Customer Login

Each of the modules was further divided in to sub components and carry out the development. Development started with the admin login and it was further decomposed in to login, user assignment, charges management, record etc.

Same principle applied to the rest of the modules.

Development was carried on module wise. Development was carried out in iterative and incremental development cycle. So each and every module was thoroughly tested and user feedback was gathered for further enhancements.

**CHAPTER - IV**

**Implementation**

**4.1 Technologies used.**

Following technologies are used in the Implementation phase.

**4.1.1 python-overview**:

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

**It is used for:**

* Web development (server-side),
* Software development,
* Mathematics,
* System scripting.

**What can Python do?**

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

**Why Python?**

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.
* Designing : HTML, CSS, JavaScript, Bootstrap

**Language features**

**Interpreted**

* + There are no separate compilation and execution steps like C and C++.
  + Directly run the program from the source code.
  + Internally, Python converts the source code into an intermediate form called byte codes which is then translated into native language of specific computer to run it.
  + No need to worry about linking and loading with libraries, etc.

**Platform Independent**

* + Python programs can be developed and executed on multiple operating system platforms.
  + Python can be used on Linux, Windows, Macintosh, Solaris and many more.

**Free and Open Source: Redistributable**

**High-level Language**

* In Python, no need to take care about low-level details such as managing the memory used by the program.

**Simple**

* + Closer to English language; Easy to Learn
  + More emphasis on the solution to the problem rather than the syntax

**Embeddable**

* + Python can be used within C/C++ program to give scripting capabilities for the program’s users.

**Robust**

* + Exceptional handling features
  + Memory management techniques in built

**Rich Library Support**

* + The Python Standard Library is varying vast.
  + Known as the “batteries included” philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.
  + Besides the standard library, there are various other high-quality libraries such as the [Python Imaging Library](http://www.pythonware.org/) which is an amazingly simple image manipulation library.

**Python language advantages and applications**

Python is a high level, interpreted and general purpose dynamic programming language that focuses on code readability. It has fewer steps when compared to Java and C. It was founded in 1991 by developer Guido Van Rossi. It is used in many organizations as it supports multiple programming paradigms. It also performs automatic memory management.

**ADVANTAGES:**

1) Presence of third-party modules

2) Extensive support libraries ( Num Py for numerical calculations, Pandas for data analytics etc)

3) Open source and community development

4) Easy to learn

5) User-friendly data structures

6) High-level language

7) Dynamically typed language (No need to mention data type based on value assigned, it takes data type)

8) Object-oriented language

9) Portable and Interactive

10) Portable across Operating systems

**APPLICATIONS:**

1) GUI based desktop applications(Games, Scientific Applications)

2) Web frameworks and applications

3) Enterprise and Business applications

4) Operating Systems

5) Language Development

6) Prototyping

**ORGANIZATION USING PYTHON:**

1) Google(Components of Google spider and Search Engine)

2) Yahoo(Maps)

3) YouTube

4) Mozilla

5) Dropbox

6) Microsoft

7) Cisco

8) Spotify

9) Quora

**Python - data visualization using bokeh**

Bokeh is a data visualization library in Python that provides high-performance interactive charts and plots. Bokeh output can be obtained in various mediums like notebook, html and server. It is possible to embed bokeh plots in Django and flask apps.

**Bokeh provides two visualization interfaces to users:**

**bokeh.models :**A low level interface that provides high flexibility to application developers.

**bokeh.plotting:** A high level interface for creating visual glyphs.

**4.1.2 DJANGO:**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

**What will you find here?**

**Testing Django Applications.**

* Testing is vital. The articles on testing will introduce you to unit and integration testing for your Django applications. You will also learn about the different packages and libraries available to assist with writing and running test suites.

**Rest Api**

* Learn how to create RESTFul APIs using the Django Rest Framework(DRF), an application used for rapidly building RESTful APIs based on Django models.

**Best practices**

* Learn Django best practices, recommended workflow, project structure and also how to avoid common pitfalls when building Django projects.

**Deployment**

* When your application is ready to leave the room and be deployed, the tutorials and articles on deployment will cover deployment options available to you and how to deploy your site to each one.

**Caching**

* Fast page loads improve the experience of visiting your site. Here you’ll learn about factors that slow web applications down and how you can boost performance by implementing caching.

**Why do you need a framework?**

To understand what Django is actually for, we need to take a closer look at the servers. The first thing is that the server needs to know that you want it to serve you a web page.

Imagine a mailbox (port) which is monitored for incoming letters (requests). This is done by a web server. The web server reads the letter and then sends a response with a webpage. But when you want to send something, you need to have some content. Django is something that helps you create the content.

**Django models**

What we want to create now is something that will store all the posts in our blog. But to be able to do that we need to talk a little bit about things called objects.

**Objects**

There is a concept in programming called object-oriented programming. The idea is that instead of writing everything as a boring sequence of programming instructions, we can model things and define how they interact with each other.

**Django views – time to create!**

A view is a place where we put the "logic" of our application. It will request information from the model you created before and pass it to a template. We'll create a template in the next chapter. Views are just Python functions that are a little bit more complicated

Views are placed in the views.py file.

Dynamic data in templates

We have different pieces in place: the Post model is defined in models.py, we have post\_list in views.py and the template added. But how will we actually make our posts appear in our HTML template? Because that is what we want to do – take some content (models saved in the database) and display it nicely in our template, right?

This is exactly what views are supposed to do: connect models and templates. In our post\_list view we will need to take the models we want to display and pass them to the template. In a view we decide what (model) will be displayed in a template.

**Django Forms**

The final thing we want to do on our website is create a nice way to add and edit blog posts. Django's admin is cool, but it is rather hard to customize and make pretty. With forms we will have absolute power over our interface – we can do almost anything we can imagine!

The nice thing about Django forms is that we can either define one from scratch or create a ModelForm which will save the result of the form to the model.

This is exactly what we want to do: we will create a form for our Post model.

Like every important part of Django, forms have their own file: forms.py.

**Django ORM and QuerySets**

**What is a QuerySet?**

A QuerySet is, in essence, a list of objects of a given Model. QuerySets allow you to read the data from the database, filter it and order it.

**4.1.3 SQLite**

SQLite is a software library that provides a relational database management system. The lite in SQLite means light weight in terms of setup, database administration, and required resource.

SQLite has the following noticeable features: self-contained, serverless, zero-configuration, transactional.

**Server less**

Normally, an RDBMS such as MySQL, Postgre SQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture.

SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.

The following diagram illustrates the SQLite server-less architecture:

**Self-Contained**

SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environments especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

SQLite is developed using ANSI-C. The source code is available as a big sqlite3.c and its header file sqlite3.h. If you want to develop an application that uses SQLite, you just need to drop these files into your project and compile it with your code.

**Zero-configuration**

Because of the serverless architecture, you don’t need to “install” SQLite before using it. There is no server process that needs to be configured, started, and stopped.

In addition, SQLite does not use any configuration files.

**Transactional**

All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.

In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

**SQLite distinctive features**

SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.

SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.

SQLite is capable of creating in-memory databases which are very fast to work with.

**Python sqlite3 module APIs**

Following are important sqlite3 module routines, which can suffice your requirement to work with SQLite database from your Python program. If you are looking for a more sophisticated application, then you can look into Python sqlite3 module's official documentation.

**sqlite3.connect (database [,time out ,other optional arguments])**

This API opens a connection to the SQLite database file. You can use ":memory:" to open a database connection to a database that resides in RAM instead of on disk. If database is opened successfully, it returns a connection object.

When a database is accessed by multiple connections, and one of the processes modifies the database, the SQLite database is locked until that transaction is committed. The timeout parameter specifies how long the connection should wait for the lock to go away until raising an exception. The default for the timeout parameter is 5.0 (five seconds).

If the given database name does not exist then this call will create the database. You can specify filename with the required path as well if you want to create a database anywhere else except in the current directory.

**connection.cursor ([cursorClass])**

This routine creates a cursor which will be used throughout of your database programming with Python. This method accepts a single optional parameter cursorClass. If supplied, this must be a custom cursor class that extends sqlite3.Cursor.

**cursor.execute (sql [, optional parameters])**

This routine executes an SQL statement. The SQL statement may be parameterized (i. e. placeholders instead of SQL literals). The sqlite3 module supports two kinds of placeholders: question marks and named placeholders (named style).

For example − cursor.execute ("insert into people values (?, ?)", (who, age))

**connection.total\_changes ()**

This routine returns the total number of database rows that have been modified, inserted, or deleted since the database connection was opened.

**connection.commit()**

This method commits the current transaction. If you don't call this method, anything you did since the last call to commit() is not visible from other database connections.

**cursor.fetchmany([size = cursor.arraysize])**

This routine fetches the next set of rows of a query result, returning a list. An empty list is returned when no more rows are available. The method tries to fetch as many rows as indicated by the size parameter.

**cursor.fetchall()**

This routine fetches all (remaining) rows of a query result, returning a list. An empty list is returned when no rows are available.

**4.1.4 HTML**

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages.

HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format.

**HTML page structure:**

The Basic structure of HTML page is given below. It contain some elements like head, title, body, … etc. These elements are used to build the blocks of web pages.

<html>: This is called HTML root element and used to wrap all the code.

<head>: Head tag contains metadata, title, page CSS etc. All the HTML elements that can be used inside the <head> element are:

* <style>
* <title>
* <base>
* <no script>
* <script>
* <meta>
* <title>

<body>: Body tag is used to enclosed all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

Features of HTML:

* It is platform independent.
* Images, video and audio can be added to a web page.
* Hypertext can be added to text.
* It is a markup language.

**Why learn HTML?**

* It is a simple markup language. Its implementation is easy.
* It is used to create a website.
* Helps in developing fundamentals about web programming.
* Boost professional career.

**Advantages:**

* HTML is used to build a websites.
* It is supported by all browsers.
* It can be integrated with other languages like CSS, JavaScript etc.

**Disadvantages:**

* HTML can create only static webpages so for dynamic web page other languages have to be used.
* Large amount of code has to be written to create a simple web page.
* Security feature is not good.

**CSS**

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

**What does CSS do**

* You can add new looks to your old HTML documents.
* You can completely change the look of your website with only a few changes in CSS code.

**Major Benefits of CSS:**

**1) Solves a big problem**

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

**2) Saves a lot of time**

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

**3) Provide more attributes**

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

**CSS Syntax**

A CSS rule set contains a selector and a declaration block.

**Selector:**

Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

**Declaration Block:**

The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

color: yellow;

font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

**Property:**

A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:**

Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

**Applications of CSS**

As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of them here:

* CSS saves time - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* Global web standards - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

**4.1.5 JAVASCRIPT**

JavaScript is an object-based scripting language which is lightweight and cross-platform.

JavaScript is not a compiled language, but it is a translated language.

The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

**What can in-browser JavaScript do?**

JavaScript's functionality depends on the environment it's running in. For example, Node.js supports a function which allows JavaScript to read and write arbitrary files, perform network requests, object-oriented, etc. The roles that JavaScript plays in both client-side (front end) and server-side (back end) development of applications can vary wildly.

In-browser JavaScript also allows you to perform webpage manipulation, interaction with the user and with the web server.

**JavaScript offer advantages like:**

* Show dynamic content based on the user profile.
* React to user's operations, like mouse clicks events, key presses or pointer movements.
* Support features like auto-validated form entries and interactive drop-down menus.
* Send requests to remote servers, Upload and download files.
* JavaScript code can also create movement and sound
* Ask questions to the users, Get and set cookies, show messages, switch browser tabs.
* Allows the data on to be stored in the local storage.

**What can't in-browser JavaScript do?**

JavaScript's capabilities in the browser are quite limited for the sake of the user's safety. It helps to prevent any unauthorized webpage from accessing private information.

**Examples of such limitations are:**

* JavaScript on a webpage may not allow you to copy, execute or read/write arbitrary files on the hard disk. It doesn't offer any access to Operating system functions.
* Many browsers allow it to work with files, but the access is very limited and only provided if the user is performing a specific action like, dropping a file into a browser window or selecting using <input> tag.
* JavaScript allows you to communicate over the net to the server where the current page came from. Although, it does not allow you to receive data from other sites/domains.

**Application of JavaScript**

**JavaScript is used to create interactive websites. It is mainly used for:**

* Client-side validation,
* Dynamic drop-down menus,
* Displaying date and time,
* Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
* Displaying clocks etc.

**What makes JavaScript unique?**

**The most important features which make JavaScript unique**

* It offers full integration with HTML/CSS.
* Simple things are done quickly without any complication or following strict rules.
* Supported by all major browsers and JavaScript is enabled by default.
  + 1. **BOOTSTRAP**

Bootstrap is the popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.

**What is Bootstrap**

* Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
* It is absolutely free to download and use.
* It is a front-end framework used for easier and faster web development.
* It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
* It can also use JavaScript plug-ins.
* It facilitates you to create responsive designs.

**Advantages of Bootstrap:**

* It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
* It facilitates users to develop a responsive website.
* It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

**What is a responsive Website?**

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

**What Bootstrap package contains**

* Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, and background.
* CSS: Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.
* Components: Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.
* JavaScript Plugins: Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.
* Customize: Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

Bootstrap Container

In Bootstrap, container is used to set the content's margins dealing with the responsive behaviors of your layout. It contains the row elements and the row elements are the container of columns (known as grid system).

**The container class is used to create boxed content.**

**Two container classes in Bootstrap:**

1. container
2. container-fluid
   1. **Software implementation process**

The software implementation method is a systematically structured approach to effectively integrate a software based service or component into the workflow of an organizational structure or an individual end-user.

**4.2.1 Overview**

A product software implementation method is a blueprint to get users and/or organizations running with a specific software product.

The method is a set of rules and views to cope with the most common issues that occur when implementing a software product: business alignment from the organizational view and acceptance from human view.

The implementation of product software, as the final link in the deployment chain of software production, is in a financial perspective of a major issue.

It is stated that the implementation of (product) software consumes up to 1/3 of the budget of a software purchase (more than hardware and software requirements together).

When the main point of the computer study program is to implement counter measures to bots and bugs.

* 1. **Application of theories in Implementation**

After the process for any hardware or software implementation project moves through the planning stage, companies should gradually test the new system until it is ready to fully go live. A pilot program, where a small test group within the company runs the system as part of a trial for a finite period of time, is often a key component to any implementation process.

Pilot project was carried out for about 1 month and following tasks were performed during the period.

**4.4 Application of theories in implementation**

Separately developed modules were deployed at the end users to experience them for certain time and feedback was gathered for further enhancements.

Parallel run period for about one week was implemented. At this time both the manual system and the automated system were executed in parallel. It was a bit complicated period for the end user, since there was only a single employee assigned for the data entry part at the gymnasium. And the situation gets worse during the peak hours where the number of customers increase.

User feedback was gathered for peak hours and non-peak hours. Also accuracy of live system was tested against the manual system. The results were analyzed critically and added in to the development cycle to finalize the product.

**CHAPTER - V**

**User Evaluation and Testing**

**5.1 Test plan**

Software was developed in iterative and incremental development model. So each and every iterate, each module was thoroughly tested to ensure the quality.

Bellow mentioned tests were done during the development phase

**5.1.1 Functional Testing**

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

**There are five steps that are involved while testing an application for functionality.**

|  |  |
| --- | --- |
| Steps | Description |
| I | The determination of the functionality that the intended application is meant to perform. |
| II | The creation of test data based on the specifications of the application. |
| III | The output based on the test data and the specifications of the application. |
| IV | The writing of test scenarios and the execution of test cases. |
| V | The comparison of actual and expected results based on the executed test cases. |

An effective testing practice was used throughout the development and hence it will make sure that the organization maintains the strictest of standards when it comes to software quality.

**Unit Testing**

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

**Integration Testing**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

**System Testing**

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards.

**System testing is important because of the following reasons:**

• System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

• The application is tested thoroughly to verify that it meets the functional and technical specifications.

• The application is tested in an environment that is very close to the production environment where the application will be deployed.

• System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

**Regression Testing**

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation. The intent of regression testing is to ensure that a change, such as a bug fix should not result in another fault being uncovered in the application.

**Regression testing is important because of the following reasons:**

• Minimize the gaps in testing when an application with changes made has to be tested.

• Testing the new changes to verify that the changes made did not affect any other area of the application.

• Mitigates risks when regression testing is performed on the application.

• Test coverage is increased without compromising timelines.

• Increase speed to market the product.

**Acceptance Testing**

This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client’s requirement. The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.

More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated. Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors, or interface gaps, but also to point out any bugs in the application that will result in system crashes or major errors in the application.

By performing acceptance tests on an application, the testing team will deduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

Acceptance by the user is high because each module has been experienced by the users and they have given their feedback the final live system to the end user

**Alpha Testing**

Unit testing, integration testing and system testing when combined together is known as alpha testing. During this phase, the following aspects will be tested in the application:

• Spelling Mistakes

• Broken Links

• Cloudy Directions

• The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

**Beta Testing**

This test is performed after alpha testing has been successfully performed. In beta testing, a sample of the intended audience tests the application. Beta testing is also known as pre-release testing. Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release. In this phase, the audience will be testing the following:

• Users will install, run the application and send their feedback to the project team.

• Typographical errors, confusing application flow, and even crashes.

• Getting the feedback, the project team can fix the problems before releasing the software to the actual users.

• The more issues you fix that solve real user problems, the higher the quality of your application will be.

• Having a higher-quality application when you release it to the general public will increase customer satisfaction.

Members of the gym were selected to do the Alpha testing and feedback was gathered. This was repeated several times till the expected standard reached.

**5.1.2 Non-Functional Testing**

This section is based upon testing an application from its non-functional attributes. Nonfunctional testing involves testing software from the requirements which are nonfunctional in nature but important such as performance, security, user interface, etc.

Some of the important and commonly used non-functional testing types are discussed below.

**Performance Testing**

**It is mostly used to identify any bottlenecks or performance issues rather than finding bugs in a software. There are different causes that contribute in lowering the performance of a software:**

• Network delay

• Client-side processing

• Database transaction processing

• Load balancing between servers

• Data rendering

**Performance testing is considered as one of the important and mandatory testing type in terms of the following aspects:**

• Speed (i.e. Response Time, data rendering and accessing)

• Capacity

• Stability

• Scalability

Performance testing can be either qualitative or quantitative and can be divided into different sub-types such as Load testing and Stress testing.

**Load Testing**

It is a process of testing the behavior of software by applying maximum load in terms of software accessing and manipulating large input data. It can be done at both normal and peak load conditions. This type of testing identifies the maximum capacity of software and its behavior at peak time.

Load testing was done at the peak hours at the gym, though could not figure out any significant delays or failures in the system.

Most of the time, load testing is performed with the help of automated tools such as Load Runner, App Loader, IBM Rational Performance Tester, Apache JMeter, Silk Performer, Visual Studio Load Test, etc.

Virtual users (VUsers) are defined in the automated testing tool and the script is executed to verify the load testing for the software. The number of users can be increased or decreased concurrently or incrementally based upon the requirements.

**Stress Testing**

Stress testing includes testing the behavior of a software under abnormal conditions. For example, it may include taking away some resources or applying a load beyond the actual load limit.

The aim of stress testing is to test the software by applying the load to the system and taking over the resources used by the software to identify the breaking point. This testing was performed by testing different scenarios such as:

• Shutdown or restart of network ports randomly

• Turning the database on or off

• Running different processes that consume resources such as CPU, memory, server, etc.

**Usability Testing**

Usability testing is a black-box technique and is used to identify any error(s) and improvements in the software by observing the users through their usage and operation.

**CHAPTER - VI**

**Future Work**

The project has been developed in a very short period of time and all efforts have been taken so that this project is very efficient in its execution there still exists some scope of improvement in our project. The following lists some of the enhancement that can be added incorporate into the project. Application of the project can be done more attractively. Database management and all maintenance module can be updated which helps the administrator. More security measures can be taken. There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider:

• Real-time Chat option for Customer and Owner, so that Customer can directly enquiry theirs Owner on any time through the Chat

•Video conversation option for Shop owner and Customer.

• Online payment

•Finger print matching for taking entry to shop.

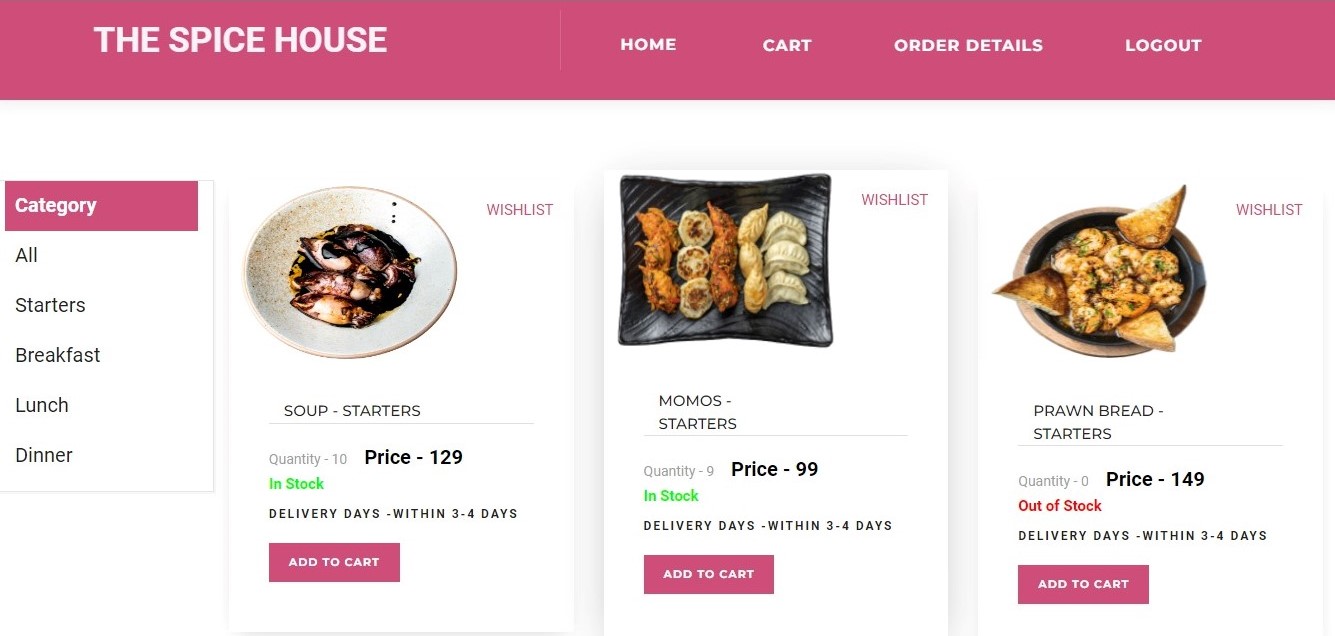
**6.1 Conclusion**

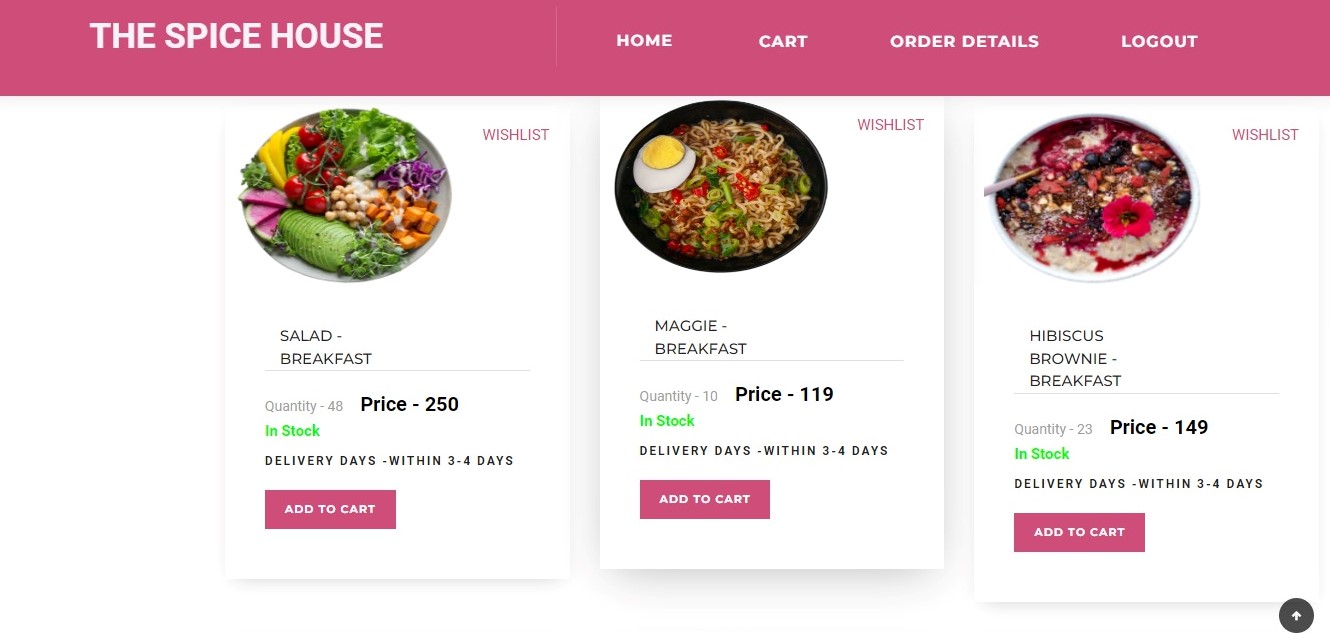
The **“Food Management System”** is successfully designed and developed to fulfilling the necessary requirements, as identified in the requirements analysis phase, such as the system is very much user friendly, form level validation and field level validation are performing very efficiently. The old manual system was suffering from a series of drawbacks. The present project has been developed to meet the aspirations indicated in the modern age.

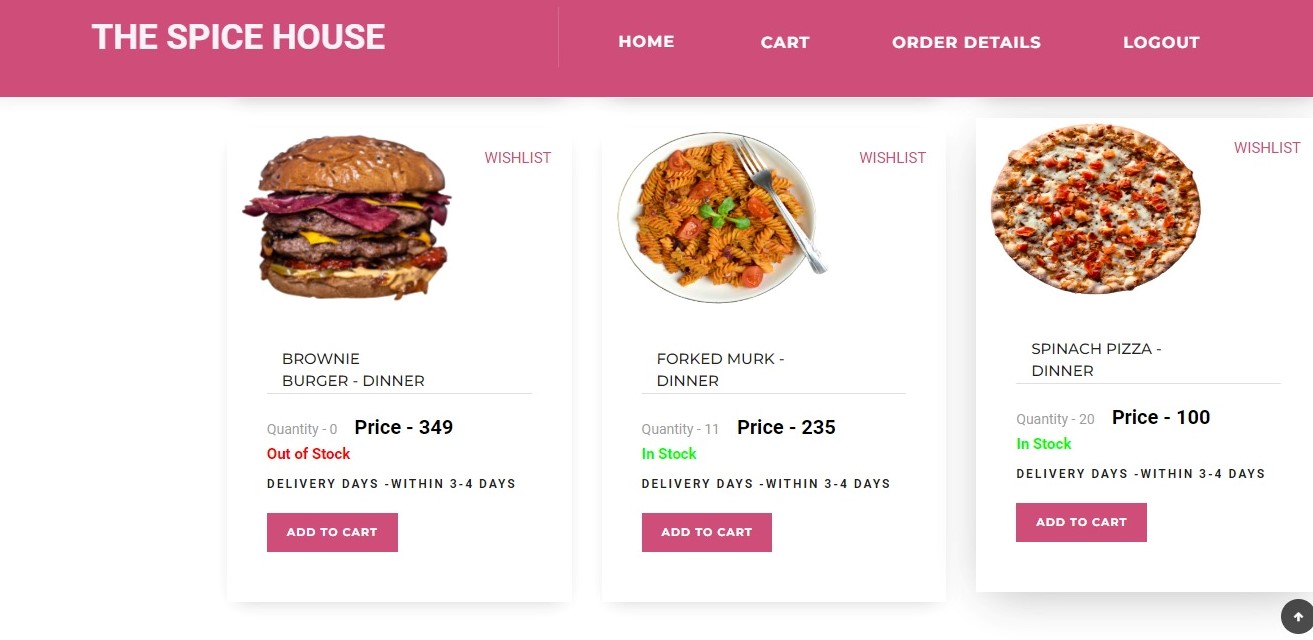
**6.2 Future Work**

**Appendix 1 (SCREENSHOTS)**

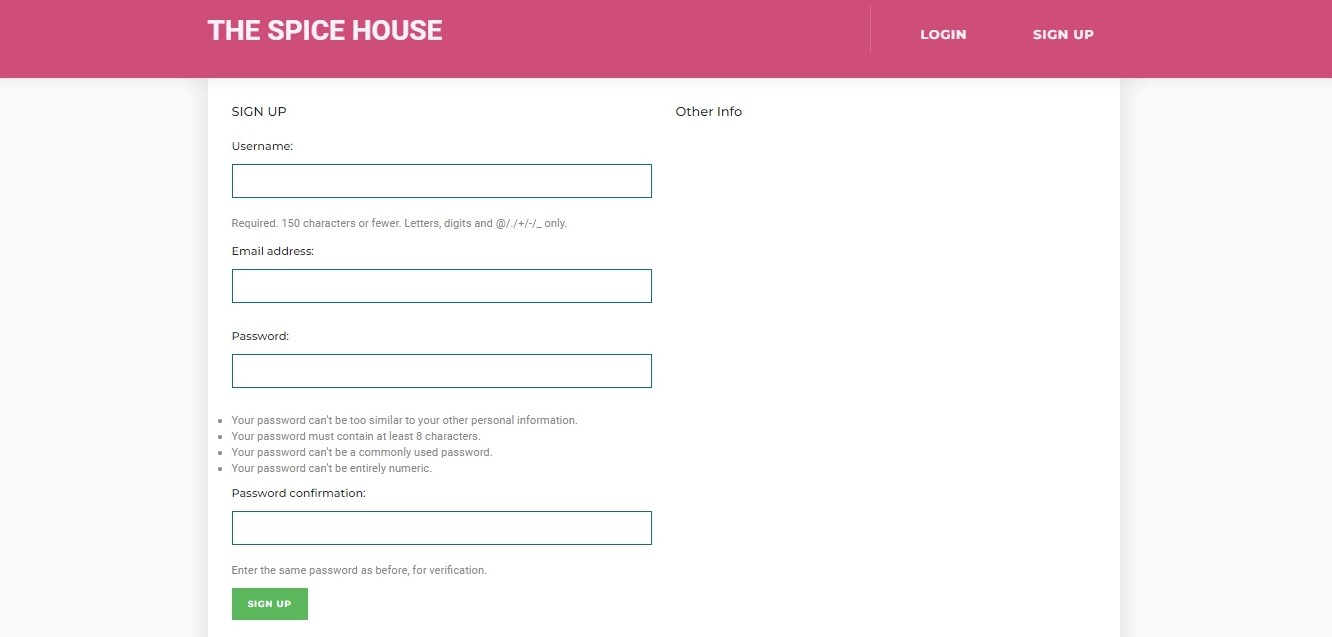
**HOME PAGE:**

****

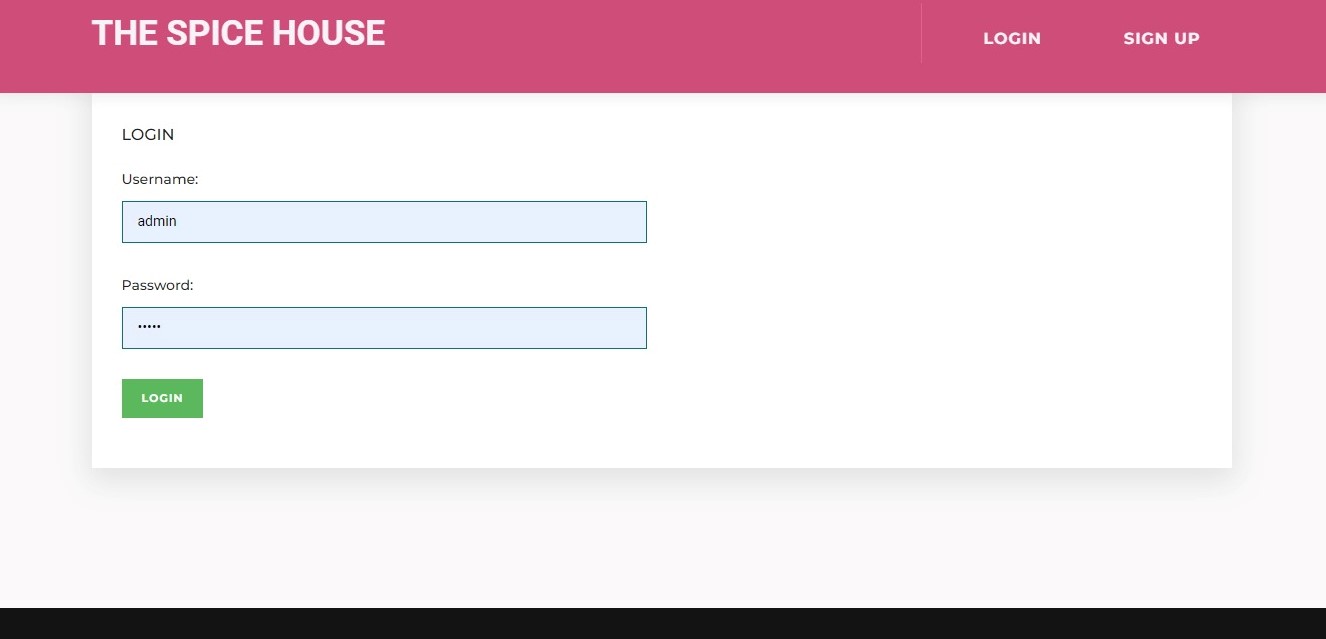
****

****

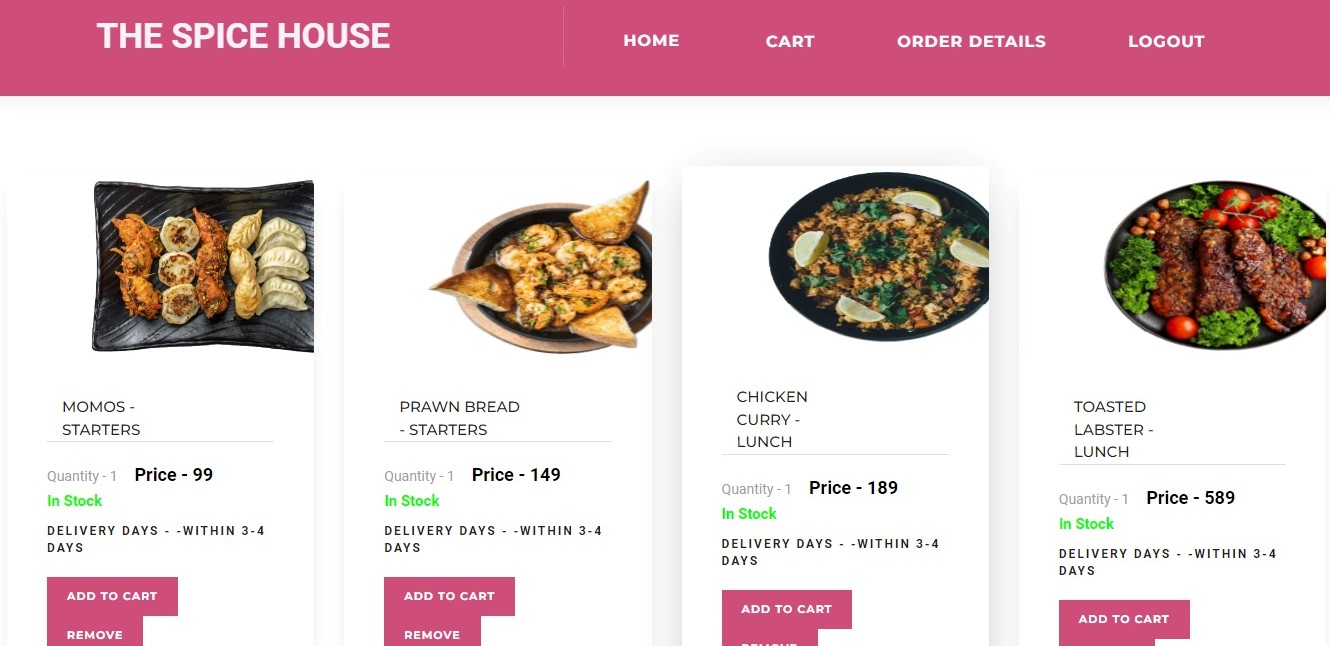
**SIGNUP FORM:**

****

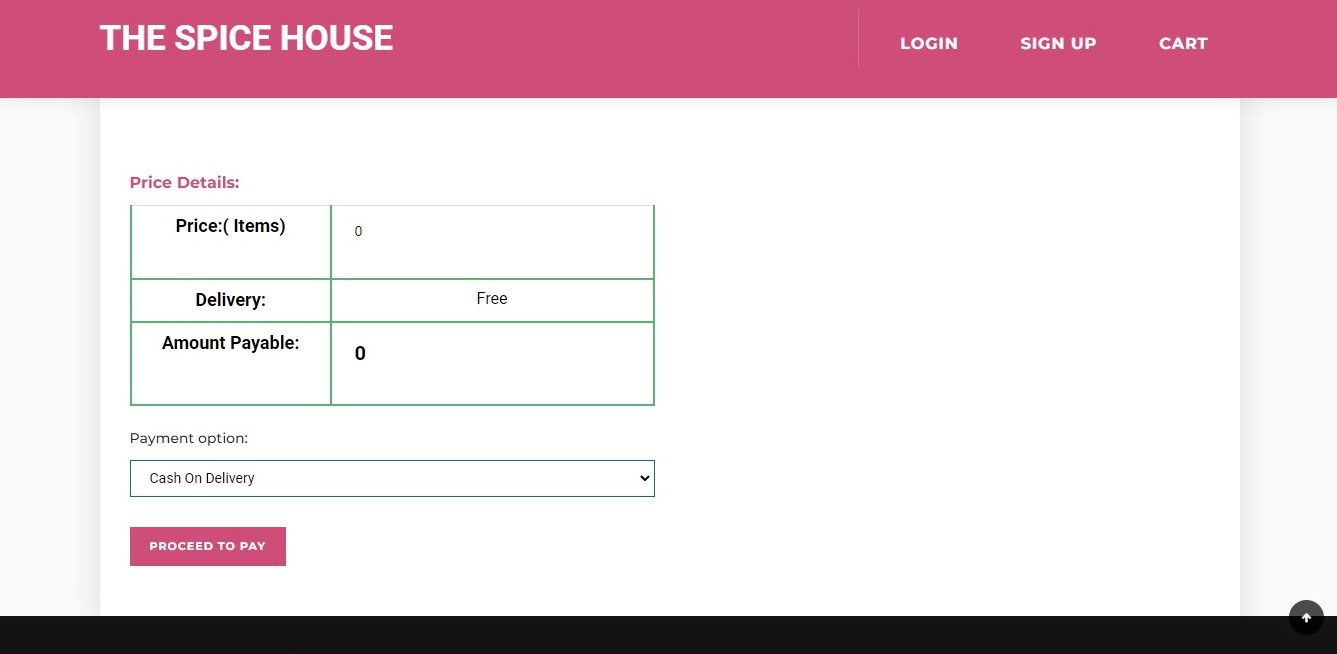
**LOGIN FORM:**

****

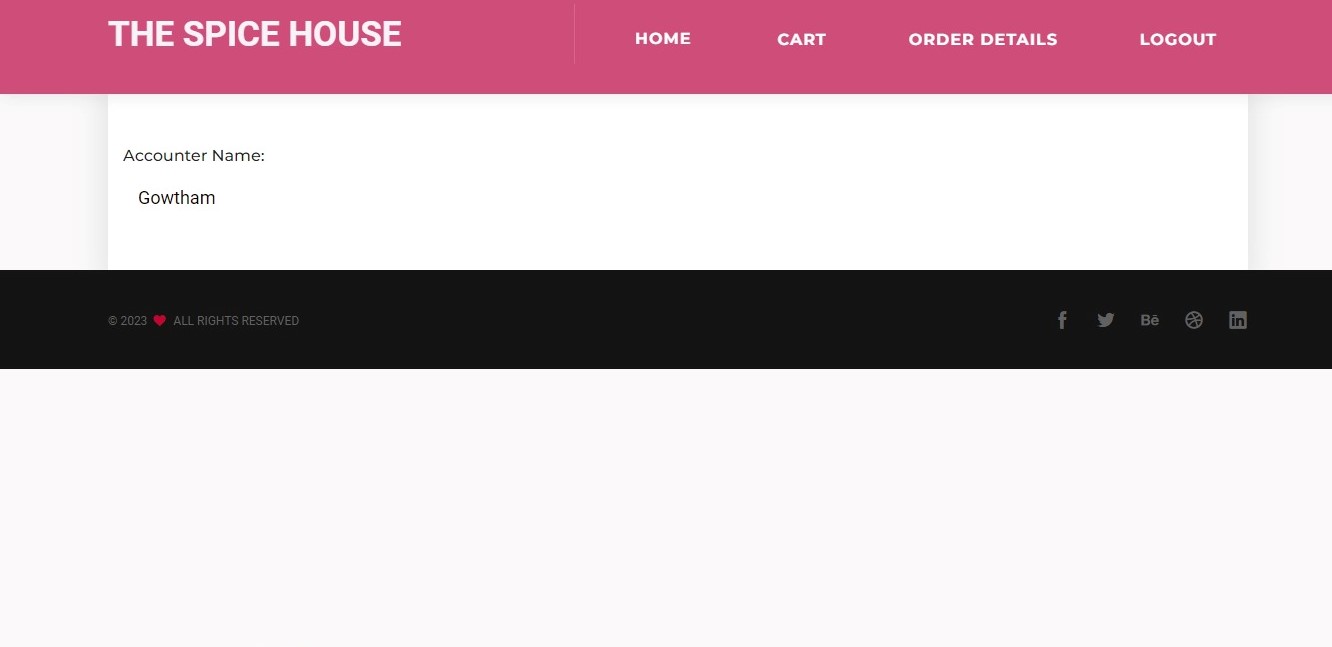
**WHISHLIST PAGE:**

****

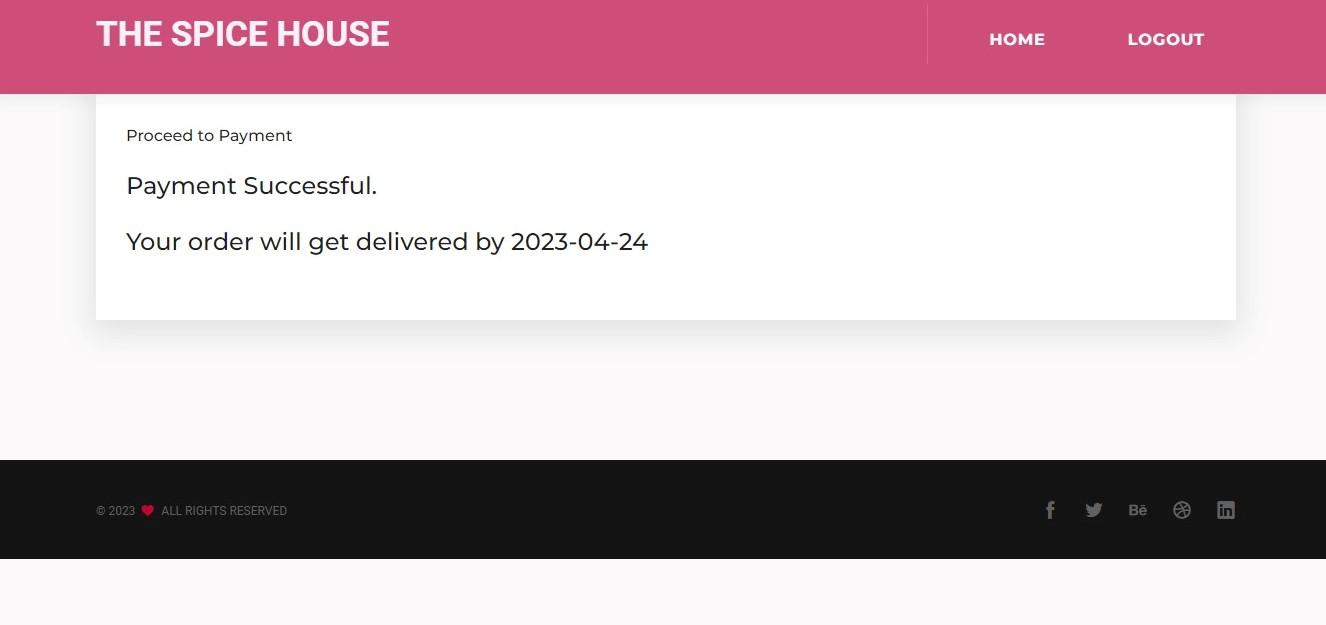
**ADD TO CART:**

****

**ORDERS DETAILS:**

****

**PAYMENT DETAILS PAGE:**

****

**Appendix 2 (SOURCE CODE):**

**Base.html:**

**{% load static %}**

**<html lang="en">**

**<!-- Mirrored from incognitothemes.com/torneo/home-classic.html by HTTrack Website Copier/3.x [XR&CO'2014], Thu, 20 Dec 2018 06:48:11 GMT -->**

**<head>**

**<meta charset="utf-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<meta name="author" content="Designsninja">**

**<!-- description -->**

**<meta name="description" content="Torneo is creative agency, corporate, simple and clean template, Based on Bootstrap responsive multipurpose agency and portfolio HTML5 template with 28 unique home page demos">**

**<!-- keywords -->**

**<meta name="keywords" content="classic, revolution slider, business, particles, portfolio, one page, bootstrap responsive, start-up, ui/ux, html5, css3, studio, branding, creative design, multipurpose, parallax, personal, masonry, grid, coming soon, carousel, career">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<title>{% block title %}organic{% endblock %}</title>**

**<link rel="shortcut icon" href="{% static 'assets/immages/favicon.ico' %}">**

**<!-- Core Style Sheets -->**

**<link rel="stylesheet" href="{% static 'assets/css/master.css' %}">**

**<!-- Responsive Style Sheets -->**

**<link rel="stylesheet" href="{% static 'assets/css/responsive.css' %}">**

**<!-- Revolution Style Sheets -->**

**<link rel="stylesheet" type="text/css" href="{% static 'revolution/css/settings.css' %}">**

**<link rel="stylesheet" type="text/css" href="{% static 'revolution/css/layers.css' %}">**

**<link rel="stylesheet" type="text/css" href="{% static 'revolution/css/navigation.css' %}">**

**</head>**

**<body>**

**<!--== Loader Start ==-->**

**<!--== Loader End ==-->**

**<!--== Wrapper Start ==-->**

**<div class="wrapper">**

**<!--== Header Start ==-->**

**<nav style="height:100px;padding:10px;background-color: #cf4e79;" class="navbar navbar-default navbar-fixed navbar white bootsnav on no-full no-border">**

**<!--== Start Top Search ==-->**

**<div class="fullscreen-search-overlay" id="search-overlay"> <a href="#" class="fullscreen-close" id="fullscreen-close-button"><i class="icofont icofont-close"></i></a>**

**<div id="fullscreen-search-wrapper">**

**<form method="get" id="fullscreen-searchform">**

**<input type="text" value="" placeholder="Type and hit Enter..." id="fullscreen-search-input" class="search-bar-top">**

**<i class="fullscreen-search-icon icofont icofont-search">**

**<input value="" type="submit">**

**</i>**

**</form>**

**</div>**

**</div>**

**<!--== End Top Search ==-->**

**<div class="container">**

**<!--== Start Atribute Navigation ==-->**

**<div class="attr-nav hidden-xs sm-display-none">**

**<ul>**

**{% if user.is\_authenticated %}**

**{% if user.is\_staff %}**

**<li class="dropdown"> <a style="background-color:transparent;color:white;font-size:16px;" href="{% url 'vo' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm" data-toggle="dropdown">Home</a>**

**<ul class="dropdown-menu">**

**<li><a href="{% url 'ao' %}">Add Products</a></li>**

**<li><a href="{% url 'vo' %}">View Products</a></li>**

**<li><a href="{% url 'payment' %}">Add Payment</a></li>**

**<li><a href="{% url 'fav' %}">View Wishlist</a></li>**

**</ul>**

**</li>**

**<li><a style="color:white;background-color:transparent;font-size:16px;" href="{% url 'logout' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">LOGOUT</a></li>**

**{% else %}**

**<li class="dropdown"> <a style="color:white;background-color:transparent;font-size:16px;border:none;" href="#" class="dropdown-toggle btn btn-xs btn-dark-outline btn-square margin-left-auto margin-right-auto display-table-sm" data-toggle="dropdown">Home</a>**

**<ul class="dropdown-menu">**

**<li><a style="color:#cf4e79;font-size:15px;" href="{% url 'vo' %}"><b>View Products</b></a></li>**

**<li><a style="color:#cf4e79;font-size:15px;" href="{% url 'fav' %}"><b>View Wishlist</b></a></li>**

**</ul>**

**</li>**

**<li><a style="color:white;background-color:transparent;font-size:16px;" href="{% url 'addcartbuy' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm"><i class="fas fa-shopping-cart"></i>Cart</a>**

**</li>**

**<li><a style="color:white;background-color:transparent;font-size:16px;" href="{% url 'order' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">ORDER DETAILS</a></li>**

**<li><a style="color:white;background-color:transparent;font-size:16px;" href="{% url 'logout' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">LOGOUT</a></li>**

**{% endif %}**

**{% else %}**

**<li><a style="color:#f7f2f7;background-color:transparent;font-size:16px;" href="{% url 'login' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">LOGIN</a></li>**

**<li><a style="color:#f7f2f7;background-color:transparent;font-size:16px;" href="{% url 'signup' %}" class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">SIGN UP</a></li>**

**<!-- <li><a style="color:#f7f2f7;background-color:transparent;font-size:16px;" href="{% url 'addcartbuy' %} " class="btn btn-md btn-info btn-square margin-left-auto margin-right-auto display-table-sm">CART</a></li> -->**

**{% endif %}**

**</ul>**

**</div>**

**<!--== End Atribute Navigation ==-->**

**<!--== Start Header Navigation ==-->**

**<div class="navbar-header">**

**<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbar-menu"> <i class="tr-icon ion-android-menu"></i> </button>**

**<div class="logo" style="font-size:35px;"> <a style="color:#f7f2f7; " "="{% url 'vo' %}"> <b>THE SPICE HOUSE</b></a>**

**</div>**

**</div>**

**<!--== End Header Navigation ==-->**

**<!--== Collect the nav links, forms, and other content for toggling ==-->**

**<div class="collapse navbar-collapse" id="navbar-menu">**

**</div>**

**<!--== /.navbar-collapse ==-->**

**</div>**

**<!-- Start Side Menu -->**

**<!-- End Side Menu -->**

**</nav>**

**<!--== Header End ==-->**

**<br>**

**<br>**

**<br>**

**{% block content %}**

**<!--== Our Services Start ==-->**

**<section>**

**<div class="container">**

**<div class="row service-box-style-03">**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="flipper">**

**<div class="text-center mb-50 main-box">**

**<div class="box-front height-300px white-bg">**

**<div class="content-wrap">**

**<i class="icofont icofont-headphone-alt font-40px dark-color"></i>**

**<h4 class="font-600">Marketing</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**<div class="box-back height-300px dark-bg">**

**<div class="content-wrap white-color">**

**<i class="icofont icofont-headphone-alt font-40px"></i>**

**<h4 class="font-600">Marketing</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="flipper">**

**<div class="text-center mb-50 main-box">**

**<div class="box-front height-300px white-bg">**

**<div class="content-wrap">**

**<i class="icofont icofont-globe-alt font-40px dark-color"></i>**

**<h4 class="font-600">Development</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**<div class="box-back height-300px dark-bg">**

**<div class="content-wrap white-color">**

**<i class="icofont icofont-globe-alt font-40px"></i>**

**<h4 class="font-600">Development</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="flipper">**

**<div class="text-center mb-50 main-box">**

**<div class="box-front height-300px white-bg">**

**<div class="content-wrap">**

**<i class="icofont icofont-magic font-40px dark-color"></i>**

**<h4 class="font-600">Design</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**<div class="box-back height-300px dark-bg">**

**<div class="content-wrap white-color">**

**<i class="icofont icofont-magic font-40px"></i>**

**<h4 class="font-600">Design</h4>**

**<p class="font-400 mt-20">Lorem Ipsum is simply dummy text of the printing and typesetting industry.</p>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Our Services End ==-->**

**<!--== Portfolio Start ==-->**

**<section class="white-bg pb-0">**

**<div class="container">**

**<div class="row">**

**<div class="col-md-12">**

**<div id="portfolio-gallery-filter" class="cbp-l-filters-alignCenter text-sm-center mb-100">**

**<div data-filter="\*" class="cbp-filter-item-active cbp-filter-item dark">**

**All**

**</div>**

**<div data-filter=".branding" class="cbp-filter-item">**

**Branding**

**</div>**

**<div data-filter=".print" class="cbp-filter-item">**

**Print Design**

**</div>**

**<div data-filter=".web-design" class="cbp-filter-item">**

**Web Design**

**</div>**

**<div data-filter=".advertising" class="cbp-filter-item">**

**Advertising**

**</div>**

**<div data-filter=".photography" class="cbp-filter-item">**

**Photography**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<div class="container-fluid">**

**<div class="row">**

**<div class="col-md-12">**

**<div id="portfolio-gallery" class="cbp">**

**<div class="cbp-item branding col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/55.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Daniel Barkle</h4>**

**<p class="default-color">Visual Identity</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item print web-design col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/56.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Versions Packaging</h4>**

**<p class="default-color">Art Direction</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item print advertising col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/57.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Juicy Redruby</h4>**

**<p class="default-color">Print Media</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item branding photography col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/58.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Fruit Up</h4>**

**<p class="default-color">Web Design</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item branding web-design col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/59.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Clini Cloud</h4>**

**<p class="default-color">Typography</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item print advertising col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/60.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Ovik Luxury</h4>**

**<p class="default-color">Branding</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item branding photography col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/61.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Loeffler Randall</h4>**

**<p class="default-color">User Interface</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**<div class="cbp-item print web-design col-md-3 col-sm-3 with-spacing">**

**<figure class="imghvr-blur text-center">**

**<img src="{% static 'assets/images/portfolio/grid/62.jpg' %}" alt="your-image">**

**<figcaption class="dark-bg">**

**<div class="center-layout">**

**<div class="v-align-middle">**

**<h4 class="mb-0 mt-10 white-color">Sofia Palmero</h4>**

**<p class="default-color">Identity</p>**

**</div>**

**</div>**

**</figcaption>**

**</figure>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Portfolio End ==-->**

**<!--== Counter Start ==-->**

**<section class="white-bg pt-80 pb-80">**

**<div class="container">**

**<div class="row counter-type-3">**

**<div class="col-md-3 col-sm-6 col-xs-12">**

**<div class="counter-wrap wow fadeInRight" data-wow-delay="0.1s">**

**<div class="pull-left">**

**<i class="icon-paintbrush default-color font-40px"></i>**

**</div>**

**<div class="pull-left pl-20">**

**<h2><span class="counter font-600">3521</span></h2>**

**<div class="line default-bg"></div>**

**<span class="dark-color text-uppercase">Happy Clients</span>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12">**

**<div class="counter-wrap wow fadeInRight" data-wow-delay="0.2s">**

**<div class="pull-left">**

**<i class="icon-speedometer default-color font-40px"></i>**

**</div>**

**<div class="pull-left pl-20">**

**<h2><span class="counter font-600">974</span></h2>**

**<div class="line default-bg"></div>**

**<h3 class="dark-color text-uppercase">Cups of Coffee</h3>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12">**

**<div class="counter-wrap wow fadeInRight" data-wow-delay="0.3s">**

**<div class="pull-left">**

**<i class="icon-genius default-color font-40px"></i>**

**</div>**

**<div class="pull-left pl-20">**

**<h2><span class="counter font-600">634</span></h2>**

**<div class="line default-bg"></div>**

**<h3 class="dark-color text-uppercase">Project Delivered</h3>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12">**

**<div class="counter-wrap wow fadeInRight" data-wow-delay="0.4s">**

**<div class="pull-left">**

**<i class="icon-strategy default-color font-40px"></i>**

**</div>**

**<div class="pull-left pl-20">**

**<h2><span class="counter font-600">449</span></h2>**

**<div class="line default-bg"></div>**

**<h3 class="dark-color text-uppercase">Awards Won</h3>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Counter End ==-->**

**<!--== Our Process Start ==-->**

**<section class="dark-bg">**

**<div class="container">**

**<div class="row our-process-style-02">**

**<div class="col-md-3 col-sm-6 col-xs-12 wow fadeInRight line xs-mb-30" data-wow-delay="0.1s">**

**<div class="icon-wrap white-bg">**

**<div class="icon">**

**<i class="icon-tools default-color font-30px"></i>**

**</div>**

**</div>**

**<div class="text-center white-color">**

**<h4 class="font-500">Design</h4>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12 wow fadeInRight line xs-mb-30" data-wow-delay="0.2s">**

**<div class="icon-wrap white-bg">**

**<div class="icon">**

**<i class="icon-globe default-color font-30px"></i>**

**</div>**

**</div>**

**<div class="text-center white-color">**

**<h4 class="font-500">Development</h4>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12 wow fadeInRight line xs-mb-30" data-wow-delay="0.3s">**

**<div class="icon-wrap white-bg">**

**<div class="icon">**

**<i class="icon-mobile default-color font-30px"></i>**

**</div>**

**</div>**

**<div class="text-center white-color">**

**<h4 class="font-500">Testing</h4>**

**</div>**

**</div>**

**<div class="col-md-3 col-sm-6 col-xs-12 wow fadeInRight line xs-mb-30" data-wow-delay="0.4s">**

**<div class="icon-wrap white-bg">**

**<div class="icon">**

**<i class="icon-browser default-color font-30px"></i>**

**</div>**

**</div>**

**<div class="text-center white-color">**

**<h4 class="font-500">Live</h4>**

**</div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Our Process End ==-->**

**<!--== Clients Start ==-->**

**<section class="white-bg">**

**<div class="container">**

**<div class="row">**

**<div class="client-slider slick">**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/1.png' %}" alt="01"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/2.png' %}" alt="02"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/3.png' %}" alt="03"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/4.png' %}" alt="04"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/5.png' %}" alt="05"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/6.png' %}" alt="06"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/7.png' %}" alt="07"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/8.png' %}" alt="08"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/9.png' %}" alt="09"/> </div>**

**<div class="client-logo"> <img class="img-responsive" src="{% static 'assets/images/clients/10.png' %}" alt="10"/> </div>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Clients End ==-->**

**<!--== Testimonails Start ==-->**

**<section>**

**<div class="overlay-bg"></div>**

**<div class="container">**

**<div class="row slick testimonial-style-2">**

**<!--== Slide ==-->**

**<div class="col-md-6 col-sm-6 col-xs-12">**

**<div class="testimonial-item">**

**<div class="testimonial-content">**

**<div class="col-md-3 text-center col-sm-3 col-xs-12 display-table">**

**<div class="v-align-middle">**

**<img class="img-responsive img-circle" src="assets/images/team/avatar-5.jpg" alt="avatar-1"/>**

**</div>**

**</div>**

**<div class="col-md-8 text-left col-sm-8 col-xs-12">**

**<h4 class="white-color mt-0 mb-0 text-uppercase font-20px xs-text-center sm-text-center">James Lange</h4>**

**<p class="default-color xs-text-center sm-text-center">Director of Media</p>**

**<h5 class="mt-10 xs-text-center sm-text-center">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sodales nec nulla ac aliquet. Duis vel nunc eget.</h5>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!--== Slide ==-->**

**<div class="col-md-6 col-sm-6 col-xs-12">**

**<div class="testimonial-item">**

**<div class="testimonial-content">**

**<div class="col-md-3 text-center col-sm-3 col-xs-12 display-table">**

**<div class="v-align-middle">**

**<img class="img-responsive img-circle" src="{% static 'assets/images/team/avatar-6.jpg' %}" alt="avatar-1"/>**

**</div>**

**</div>**

**<div class="col-md-8 text-left col-sm-8 col-xs-12">**

**<h4 class="white-color mt-0 mb-0 text-uppercase font-20px xs-text-center sm-text-center">Rachel Cory</h4>**

**<p class="default-color xs-text-center sm-text-center">Photography</p>**

**<h5 class="mt-10 xs-text-center sm-text-center">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sodales nec nulla ac aliquet. Duis vel nunc eget.</h5>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!--== Slide ==-->**

**<div class="col-md-6 col-sm-6 col-xs-12">**

**<div class="testimonial-item">**

**<div class="testimonial-content">**

**<div class="col-md-3 text-center col-sm-3 col-xs-12 display-table">**

**<div class="v-align-middle">**

**<img class="img-responsive img-circle" src="{% static 'assets/images/team/avatar-7.jpg' %}" alt="avatar-1"/>**

**</div>**

**</div>**

**<div class="col-md-8 text-left col-sm-8 col-xs-12">**

**<h4 class="white-color mt-0 mb-0 text-uppercase font-20px xs-text-center sm-text-center">Thomas Peter</h4>**

**<p class="default-color xs-text-center sm-text-center">PHP Developer</p>**

**<h5 class="mt-10 xs-text-center sm-text-center">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sodales nec nulla ac aliquet. Duis vel nunc eget.</h5>**

**</div>**

**</div>**

**</div>**

**</div>**

**<!--== Slide ==-->**

**<div class="col-md-6 col-sm-6 col-xs-12">**

**<div class="testimonial-item">**

**<div class="testimonial-content">**

**<div class="col-md-3 text-center col-sm-3 col-xs-12 display-table">**

**<div class="v-align-middle">**

**<img class="img-responsive img-circle" src="{% static 'assets/images/team/avatar-8.jpg' %}" alt="avatar-1"/>**

**</div>**

**</div>**

**<div class="col-md-8 text-left col-sm-8 col-xs-12">**

**<h4 class="white-color mt-0 mb-0 text-uppercase font-20px xs-text-center sm-text-center">Isla Mack</h4>**

**<p class="default-color xs-text-center sm-text-center">Front-end Ninja</p>**

**<h5 class="mt-10 xs-text-center sm-text-center">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sodales nec nulla ac aliquet. Duis vel nunc eget.</h5>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**</div>**

**<video autoplay="" muted="" loop="" controls="" class="html5-video">**

**<source src="{% static 'assets/videos/explore.mp4' %}" type="video/mp4">**

**<source src="{% static 'assets/videos/explore.webm' %}" type="video/webm">**

**</video>**

**</section>**

**<!--== Testimonails End ==-->**

**<!--== Blog Posts Start ==-->**

**<section class="grey-bg">**

**<div class="container">**

**<div class="row">**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="post">**

**<div class="post-img"> <img class="img-responsive" src="{% static 'assets/images/post/post-01.jpg' %}" alt=""/> </div>**

**<div class="post-info all-padding-40">**

**<h6>January 15, 2018</h6>**

**<h3><a href="blog-grid.html">How These Different Book Covers Reflect the Design</a></h3>**

**<hr>**

**<p class="mt-10"> <span> <a class="extras-wrap" href="#"><i class="icofont icofont-chat"></i><span>5 Comments</span></a> <span class="extras-wrap"><i class="icofont icofont-clock-time"></i><span>5 Minutes</span></span> </span> </p>**

**<a class="readmore" href="#"><span>Read More</span></a> </div>**

**</div>**

**</div>**

**<!--== Post End ==-->**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="post">**

**<div class="post-img"> <img class="img-responsive" src="{% static 'assets/images/post/post-02.jpg' %}" alt=""/> </div>**

**<div class="post-info all-padding-40">**

**<h6>April 19, 2014</h6>**

**<h3><a href="blog-grid.html">8 Colorful Toys Designed to Spark the Imagination</a></h3>**

**<hr>**

**<p class="mt-10"> <span> <a class="extras-wrap" href="#"><i class="icofont icofont-chat"></i><span>5 Comments</span></a> <span class="extras-wrap"><i class="icofont icofont-clock-time"></i><span>8 Minutes</span></span> </span> </p>**

**<a class="readmore" href="#"><span>Read More</span></a> </div>**

**</div>**

**</div>**

**<!--== Post End ==-->**

**<div class="col-md-4 col-sm-4 col-xs-12">**

**<div class="post">**

**<div class="post-img"> <img class="img-responsive" src="{% static 'assets/images/post/post-03.jpg' %}" alt=""/> </div>**

**<div class="post-info all-padding-40">**

**<h6>Febuary 13, 2018</h6>**

**<h3><a href="blog-grid.html">User Experience Design is one of the most rapidly developing</a></h3>**

**<hr>**

**<p class="mt-10"> <span> <a class="extras-wrap" href="#"><i class="icofont icofont-chat"></i><span>5 Comments</span></a> <span class="extras-wrap"><i class="icofont icofont-clock-time"></i><span>Just Now</span></span> </span> </p>**

**<a class="readmore" href="#"><span>Read More</span></a> </div>**

**</div>**

**</div>**

**<!--== Post End ==-->**

**</div>**

**</div>**

**</section>**

**<!--== Blog Posts End ==-->**

**<!--== Call to Action Start ==-->**

**<section class="parallax-bg fixed-bg pt-80 pb-80" data-parallax-bg-image="{% static 'assets/images/background/parallax-bg-6.jpg' %}" data-parallax-speed="0.8" data-parallax-direction="up">**

**<div class="parallax-overlay"></div>**

**<div class="container">**

**<div class="row">**

**<div class="col-md-12 col-sm-12 centerize-col text-center white-color relative">**

**<span class="sm-mb-30 font-20px display-inline-block mr-30">No Coding required for build your page. Torneo delivers everything.</span>**

**<a class="btn btn-md btn-light-outline btn-square">Purchase Now for $14</a>**

**</div>**

**</div>**

**</div>**

**</section>**

**<!--== Call to Action End ==-->**

**{% endblock %}**

**<!--== Footer Start ==-->**

**<footer class="footer dark-block">**

**<div class="footer-copyright">**

**<div class="container">**

**<div class="row">**

**<div class="col-md-6 col-xs-12">**

**<div class="copy-right text-left">© 2023<i class="icon icofont icofont-heart-alt"></i> All rights reserved</div>**

**</div>**

**<div class="col-md-6 col-xs-12">**

**<ul class="social-media">**

**<li><a href="#" class="icofont icofont-social-facebook"></a></li>**

**<li><a href="#" class="icofont icofont-social-twitter"></a></li>**

**<li><a href="#" class="icofont icofont-social-behance"></a></li>**

**<li><a href="#" class="icofont icofont-social-dribble"></a></li>**

**<li><a href="#" class="icofont icofont-social-linkedin"></a></li>**

**</ul>**

**</div>**

**</div>**

**</div>**

**</div>**

**</footer>**

**<!--== Footer End ==-->**

**<!--== Go to Top ==-->**

**<a href="javascript:" id="return-to-top"><i class="icofont icofont-arrow-up"></i></a>**

**<!--== Go to Top End ==-->**

**</div>**

**<!--== Wrapper End ==-->**

**<!--== Javascript Plugins ==-->**

**<script src="{% static 'assets/js/jquery.min.js' %}"></script>**

**<script src="{% static 'assets/js/smoothscroll.js' %}"></script>**

**<script src="{% static 'assets/js/plugins.js' %}"></script>**

**<script src="{% static 'assets/js/master.js' %}"></script>**

**<!-- Revolution js Files -->**

**<script type="text/javascript" src="{% static 'revolution/js/jquery.themepunch.tools.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/jquery.themepunch.revolution.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.actions.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.carousel.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.kenburn.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.layeranimation.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.migration.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.navigation.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.parallax.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.slideanims.min.js' %}"></script>**

**<script type="text/javascript" src="{% static 'revolution/js/revolution.extension.video.min.js' %}"></script>**

**<!--== Javascript Plugins End ==-->**

**<script type="text/javascript">**

**var priceval=new Array();**

**var defaultval=new Array();**

**var sum=0**

**var pricename=new Array();**

**for (i = 1; i <= $("#idcount").val(); i++) {**

**console.log(i);**

**pricename[i]="#bu\_price"+i**

**console.log(pricename[i]);**

**priceval[i]=parseInt($(pricename[i]).val());**

**defaultval[i]=parseInt($(pricename[i]).val());**

**sum=sum+priceval[i]**

**console.log(priceval[i])**

**$("#buy\_price").val(sum);**

**$("#buy\_total").val(sum)**

**}**

**function changequty(quty,count,id){**

**// alert(price);**

**console.log(id)**

**var tot=0;**

**var i;**

**for (i = 1; i <= count; i++) {**

**if(i==id){**

**//console.log(pricename[i]+"");**

**$(pricename[i]).val((quty\*defaultval[i]))**

**priceval[i]=quty\*defaultval[i];**

**console.log(priceval[i])**

**}**

**tot=tot+priceval[i];**

**$("#buy\_price").val(tot);**

**$("#buy\_total").val(tot)**

**}**

**}**

**function addtocart(id,price)**

**{**

**//alert('hai');**

**$.ajax({**

**url: '/cart/add\_to\_cart\_album',**

**data : {**

**'price': price,**

**'id':id**

**},**

**success: function (data) {**

**console.log("Added succesfully")**

**}**

**});**

**}**

**function addwishlist(id,price)**

**{**

**//alert('hai');**

**$.ajax({**

**url: '/cart/wishlist\_album',**

**data : {**

**'price': price,**

**'id':id**

**},**

**success: function (data) {**

**console.log("Added succesfully")**

**}**

**});**

**}**

**</script>**

**</body>**

**<!-- Mirrored from incognitothemes.com/torneo/home-classic.html by HTTrack Website Copier/3.x [XR&CO'2014], Thu, 20 Dec 2018 06:51:51 GMT -->**

**</html>**

**view.py🡪**

**def vegetable(request):**

**objects\_list=organicfood.objects.filter(category="STARTERS")**

**#print("vegetable")**

**#print(objects\_list)**

**return render(request,'vo.html',{"object\_list":objects\_list})**

**def fruits(request):**

**objects\_list=organicfood.objects.filter(category="BREAKFAST")**

**#print("fruits")**

**return render(request,'vo.html',{"object\_list":objects\_list})**

**def lunch(request):**

**objects\_list=organicfood.objects.filter(category="LUNCH")**

**print("sprouts")**

**return render(request,'vo.html',{"object\_list":objects\_list})**

**def dinner(request):**

**objects\_list=organicfood.objects.filter(category="DINNER")**

**#print("dinner")**

**return render(request,'vo.html',{"object\_list":objects\_list})**

**urls.py 🡪**

**from django.urls import path**

**from . import views**

**urlpatterns=[**

**path('',views.ao,name="ao"),**

**path('foods/',views.vo,name="foods"),**

**path('starters/',views.vegetable,name="starters"),**

**path('breakfast/',views.fruits,name="bf"),**

**path('lunch/',views.lunch,name="lunch"),**

**path('dinner/',views.dinner,name="dinner")**

**models.py 🡪**

**from django.db import models**

**# Create your models here.**

**class organicfood(models.Model):**

**foodname=models.CharField(max\_length=100)**

**category=models.CharField(max\_length=300)**

**quantity=models.IntegerField(default=0)**

**price=models.IntegerField(default=0)**

**productimg=models.ImageField()**

**is\_favorite = models.BooleanField(default=False)**