**Online Pizza Ordering System**

**Project Synopsis**

**Version 2.0**

Course Name (Course Code):-

Industrial Training(ECS899)

Degree:-

**BACHELOR OF TECHNOLOGY (CSE) VIIIth semester**



**COLLEGE OF COMPUTING SCIENCES AND INFORMATION TECHNOLOGY**

**TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD**

|  |  |
| --- | --- |
| PROJECT GUIDE;  Mr. Ankur Sisodia | SUBMITTED BY:  **Student Name : Prakhar Jain**  **Student id : TCA1609077**  **Student Name : Mohfeez Ahmad**  **Student id : TCA1711005**  **Student Name : Sarang Vaidya**  **Student id : TCA1711011** |

**INDEX**

* INTRODUCTION ABOUT PROJECT…………..………………………….…1
* INTRODUCTION TO PYTHON..………………………………………….…..2
* IMPLEMENTATION DETAILS………………………..………………….…..3
* PROJECT DISCUSSION …..…………………………………………….…….4
* FUTURE IMPLICATIONS .……….…………………………………….…….5
* CONCLUSION ………………………………………………………………...6
* FIGURES ………………………………..……………………………………..7
* REFERRENCES ………..……………………………………………………...8

# INTRODUCTION ABOUT PROJECT

Online Pizza ordering system is a web-based application which enables customers to order their pizzas online for home delivery. As the internet users are increasing exponentially, there is need of Online Pizza ordering system for taking orders from customers. This system not only improves customer’s experience but also eases the workload on the staff .

As there is no visual menu shown during a phone call, the employees have to repeat a lot of things again and again to the customers. It’s a time-consuming process which at times irritates customers. Also, it takes a lot of time of the staff. It would be much more comfortable for the customers to have an online pizza ordering system. It would be hassle-free for users as they can select the pizzas they want and make payment for it. Also, it will reduce the purchasing time for customers.

This system will help customers in ordering pizzas online and customer satisfaction will be more. This system will provide customer all the details of his order before making the order. This confirmation will help customers to check the items ordered with their prices.

As most of the things will be performed online, it will reduce the usage of paper. This system will make things easier for staff as whole ordering process is done by the customer only. The menu will be visible to the customer with the pizzas and other non-pizza products on offer. All the ingredients will be shown at their prices. Now the customer chooses their pizza and make and select the quantity for it. Our aim is to make an ordering system that separates ordering pizzas from ordering the side dishes (non-pizza products) in an intuitive way. The ordering system provides the user with two tabs: one for side orders, and one for delivery details. Customers can switch to any tab any time. Hence the system will decrease workload of the employees and provides benefit due to the database / information system. Information will be stored in the system and can be viewed at any time. The system will be able to guide a user through the website and make then complete their pizza order. When they are done with filling in all information regarding their order they can complete the order to send it to the Pizza Palace.

The website runs on Open Source software, it is written in Python and uses MySQL as the database manager. The Pizza Palace aims to increase the sales, due to the easy to use order online website. The system will give the employees more time to “work” rather than to accept orders by phone. The focus is to create an “easy to use” website, which will allow a first time customer to complete their order with ease.

# INTRODUCTION TO PYTHON

## What is Python?

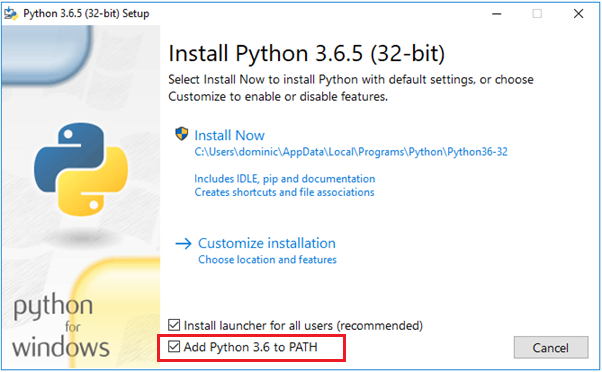
* Python is an interpreted high-level programming language. It has advantages of both scripting and programming languages.
* It is very useful for rapid application development
* It is easy to learn when comparing with other programming languages and has a design philosophy that emphasizes code readability, mainly using indentation.
* The language was originally created by Guido van Rossum and first released on 1991.
* Python programming language has a wide range of applications from Web Development, scientific and mathematical computing.
* Nowadays, Python is gaining more attention as it's great for data analysis, artificial intelligence and scientific computing
* It does not need a compiler to run the application. It’s basically an interpreter language.

## Few IDE for Python Development

* IDLE (Integrated Development Environment for Python)
  + This is the default IDE available with Python installation. In this tutorial, I am running my example programs only on IDLE. There are other IDEs as well which have lots of features. I will give details about those IDEs in upcoming parts.
* Visual Studio
* PyCharm
* Anaconda
* Tkinter

## How to Install/Configure Python in a Windows Machine

* You can download the latest version of python from the below URLs:
  + <https://www.python.org/ftp/python/3.6.5/python-3.6.5.exe>
  + <https://www.python.org/downloads/>
* Here, I am detailing about the installation and configuration in Windows 10 operating system.
* Execute Python-3.6.5.exe

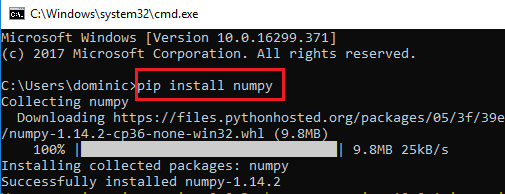


* By default, ‘Add Python 3.6 to Path’ check box is unchecked, but I suggest you check that box. This will set the environment variables automatically. So we can directly execute ‘Python’ from the windows command prompt.

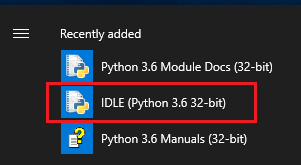
Now open the windows command prompt:

Type ‘Python’ in command prompt and it will give you the python prompt where you can type and execute Python commands.  
As you can see, I was able to print "Welcome to python programming".

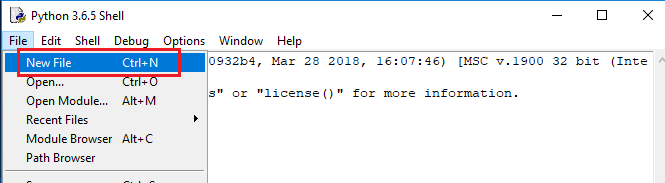
If you want to install a different python package which is not available with the default python installation, you can use the ‘Pip’ command.



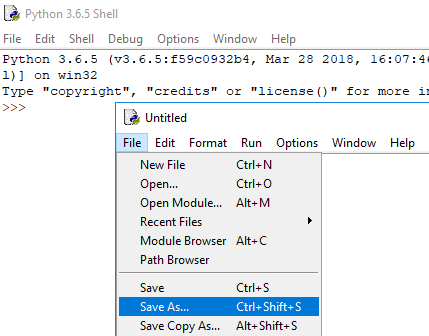
Now we can move on to the default IDE which is available with python installation.



‘Python 3.6.5 Shell’ will get opened and then click on ‘New File’



A new window will get opened and I saved it as ‘Prgoram1.py’ and here we can write programs. When we click on ‘F5’ or ‘Run -> Run Module’ window, the program will execute and provide the output in the output window.



**Now let us directly dive into the sample programs.**

## Keywords in Python

['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif',

'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',

'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

To get the list of keywords, run the below code:

|  |
| --- |
| Input |
| https://www.codeproject.com/KB/tips/1240791/Input-1.png |
| Output |
| https://www.codeproject.com/KB/tips/1240791/Output-1.png |

Here, we are importing the ‘keyword’ library file. We can see the list of library files in default Python installation location

## Arithmetic Operators

|  |  |
| --- | --- |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus - Divides left hand operand by right hand operand and returns remainder |
| \*\* | Exponent - Performs exponential (power) calculation on operators |
| // | Floor Division – The division of operands where the result is the quotient in which the digits after the decimal point are removed. |

|  |
| --- |
| Input |
| https://www.codeproject.com/KB/tips/1240791/Input-2.png |
| Output |
| https://www.codeproject.com/KB/tips/1240791/Output-2.png |

There are few things to notice in the above example other than the use of operators.

For adding comments in the code, we use the hash (#) symbol. For multiline comments, we can use triple quotes, either ''' or " " ".

There are other operators available,

For example: Comparison Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators and Identify Operators.

I am not going into the details of each of the operators, you can refer to the below link to get more information about the different operators:

* <https://www.tutorialspoint.com/python/python_basic_operators.htm>

To receive inputs from the key board, we can use ‘input’ keyword.

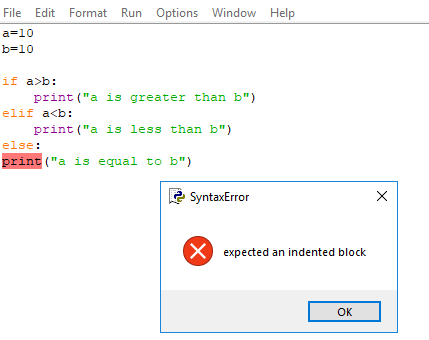
|  |
| --- |
| Input |
| https://www.codeproject.com/KB/tips/1240791/Keyboard-in.png |
| Output |
| https://www.codeproject.com/KB/tips/1240791/Keyboard-ouput.png |

## If statements

|  |
| --- |
| Input |
| https://www.codeproject.com/KB/tips/1240791/if-input-1.PNG |
| Output |
| https://www.codeproject.com/KB/tips/1240791/Output-3.png |

In the above examples, we can see there is colon ( :) after each statement. Also make sure the **indentation is the same**. There is no elseif, instead in Python we are using ‘elif’ keyword.

If the indentation is not the same in the above code, it will throw the below error. So we should be very careful about these indentations while doing programming.



## For loop

I will give some examples of ‘for loop’ in python. And we can read and understand the logic like plain English.

**Example 1:**

|  |
| --- |
| **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex1-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex1-out.png |

**Example 2:**

|  |
| --- |
| **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex2-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex2-out.png |

In this example, print () comes with a parameter called ‘end’. By default the value of this parameter is ‘\n’. But we can end a print statement with any character/string using the parameter ‘end’. Here I included a space at the end of the each print statement.

**Example3:**

|  |
| --- |
| **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex3-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex3-out.png |

In the above example, we are looping through a range of number. We can specify the start and end. It will print up to (n-1), if n is the upper range. Here all parameters in range () should be integers (either positive or negative).

**Example 4:**

|  |
| --- |
| **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex4-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex4-out.png |

Here the third parameter in range () function represents the stepping number. I.E In the above example, the difference of each number in the printed list is 2.

## For loop - with else

|  |
| --- |
| Python supports to have an else statement associated with a loop statement.  **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex5A-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex5A-out.png |

|  |
| --- |
| **Input** |
| https://www.codeproject.com/KB/tips/1240791/ex5B-in.png |
| **Output** |
| https://www.codeproject.com/KB/tips/1240791/ex5B-out.png |

If the else statement is used with a for loop, the else statement is executed when the loop has finished iterating the list.A break statement can be used to stop a for loop. In this case, the else part is ignored.  Always make sure the indentation is correct otherwise it will give you unexpected results.

## Data Types

Python has the following standard data types:

* Numbers
* String
* List
* Tuple
* Set
* Dictionary

# Implementation Details

* 1. ***: The Opening Page-*** This is the very first page that opens up as the code is run. It asks the customer for their username, and password. If a customer already has an account made then after entering his/her username and password he/she can press the Login button which would direct them to the main window where they can see the entire menu. Otherwise if the customer doesn’t have an existing account they can click the register button and that will direct them to the registration window where they can register themselves and then come back to the opening page to Log in. If the customer who doesn’t have any existing account tries to log in by putting some random entries then the page gives the message that they need to register first.

In case if the customer is entering a wrong password the message pops up to enter a correct password and if the customer forgets his/her password the option to recover password is also available. Once the user clicks the ‘ok’ button of the message box that shows up, ‘recover my password’ button pops up clicking which one will be asked to give the e-mail address they want the password to be sent to. After this the new password is sent to the e-mail address the user provides and the notification is shown that the password has been sent, now the user can use it to log into the website. Also if the password couldn’t be sent then there must be some internet connection problem and the user is notified of the same through a message telling that the password couldn’t be sent and the user need to check their internet connection. The coding concerned with this page is as follows-

sf.scr.configure(bg='white') sf.f=Frame(sf.scr,bg='white') sf.f.place(x=0,y=0,width=1900,height=900) sf.canvas=Canvas(sf.f,bg='PeachPuff',bd=-2) sf.canvas.place(x=0,y=36,width=1900,height=705)

sf.img=ImageTk.PhotoImage(Image.open('background\_.jpg')) sf.canvas.create\_image(681,405,image=sf.img) sf.l.append(sf.canvas) sf.username=Label(sf.f,text='Username') sf.username.place(x=300,y=130,width=110,height=25) sf.l.append(sf.username)

sf.un\_entry=Entry(sf.f,bg='azure')

sf.un\_entry.place(x=440,y=130,width=200,height=25) sf.l.append(sf.un\_entry) sf.passw=Label(sf.f,text='Password') sf.passw.place(x=300,y=175,width=110,height=25) sf.l.append(sf.passw) sf.p\_entry=Entry(sf.f,show="\*",bg='azure') sf.p\_entry.place(x=440,y=175,width=200,height=25) sf.l.append(sf.p\_entry)

sf.submit=Button(sf.f,text='LOGIN',command=lambda:sf.result("login")) sf.submit.place(x=300,y=230,width=100,height=28) sf.l.append(sf.submit) sf.r=Button(sf.f,text='REGISTER',command=sf.register) sf.r.place(x=440,y=230,width=130,height=28)

sf.l.append(sf.r) def result(sf,val): if val=="login":

if not len(sf.un\_entry.get()) or not len(sf.p\_entry.get()):

messagebox.showinfo("Invalid credentials","Please fill both the fields to continue.\nPlease try again.”)

else:

x=sf.cur.execute("select count(\*) from staff where user=%r"%(sf.un\_entry.get())) if list(x)[0][0]==0:

messagebox.showinfo("Invalid credentials.","Username %r doesn't exist.\nPlease 'register' to continue."%(sf.un\_entry.get()))

sf. init () else:

sf.MAIL=list(sf.cur.execute("select email from staff where user=%r"%(sf.un\_entry.get())))[0][0] sf.NAME=list(sf.cur.execute("select name from staff where user=%r"%(sf.un\_entry.get())))[0][0] print(sf.NAME)

x=sf.cur.execute("select count(\*) from staff where passw=%r"%(sf.p\_entry.get())) if list(x)[0][0]:

sf.create\_order('1') else:

messagebox.showinfo("Wrong password","Please enter a valid password\nForgot password ?") sf.rbutton=Button(sf.scr,text='Recover my password',command=sf.recover\_password) sf.rbutton.place(x=720,y=210,width=130,height=30)

sf.l.append(sf.rbutton)

* 1. ***: The Registration Page-*** This page registers the users or customers. It asks for details like name, username, password and E-mail, after the users are done entering these, they can click ‘Register Me’ to register themselves. If all the entries are not filled then a message pops up asking the user to fill all the entries. So once that is done, the application makes sure that the username entered by the customer doesn’t exist already and if it does then a message is shown that the user name already exists and so the customer need to enter some other username. In case of password, the application ensures that the password and retype password entries are the same and if not then a message is shown to the customers asking them to keep both entries same. While entering the E-mail address, a proper e-mail address should be entered otherwise ‘invalid e-mail address’ message pops up. Thus after all entries are filled correctly the message that the user has been registered successfully pops up. Now the users are re-directed to the Opening page where using the username and password they have registered with they can Log into the Pizza Palace Site. The code related to this page is as follows-

def register(sf): try:

sf.flush() except: try: sf.flush() except:

Pass

sf.f.config(bg='medium spring green'

sf.canvas=Canvas(sf.f,bg='PeachPuff',bd=-2) sf.canvas.place(x=0,y=0,width=1900,height=850) sf.img=ImageTk.PhotoImage(Image.open('background.jpg')) sf.canvas.create\_image(681,405,image=sf.img) sf.l.append(sf.canvas)

sf.Name=Label(sf.f,text='Name') sf.Name.place(x=400,y=100,width=110,height=25) sf.l.append(sf.Name)

sf.N\_entry=Entry(sf.f) sf.N\_entry.place(x=540,y=100,width=200,height=25) sf.l.append(sf.N\_entry) sf.name=Label(sf.f,text='Userame') sf.name.place(x=400,y=140,width=110,height=25) sf.l.append(sf.name)

sf.n\_entry=Entry(sf.f) sf.n\_entry.place(x=540,y=140,width=200,height=25) sf.l.append(sf.n\_entry) sf.plabel=Label(sf.f,text='Password') sf.plabel.place(x=400,y=180,width=110,height=25) sf.l.append(sf.plabel) sf.p\_entry=Entry(sf.f,show="\*") sf.p\_entry.place(x=540,y=180,width=200,height=25) sf.l.append(sf.p\_entry) sf.plabel2=Label(sf.f,text='Retype password') sf.plabel2.place(x=400,y=220,width=110,height=25) sf.l.append(sf.plabel2) sf.p\_entry2=Entry(sf.f,show="\*")

sf.p\_entry2.place(x=540,y=220,width=200,height=25) sf.l.append(sf.p\_entry2)

sf.e\_mail=Label(sf.f,text="Email") sf.e\_mail.place(x=400,y=260,width=110,height=25) sf.l.append(sf.e\_mail)

sf.e\_entry=Entry(sf.f) sf.e\_entry.place(x=540,y=260,width=200,height=25) sf.l.append(sf.e\_entry)

def result(sf,val): elif val=="register":

if not len(sf.n\_entry.get()) or not len(sf.N\_entry.get()) or not len(sf.p\_entry.get()) or not len(sf.p\_entry2.get()) or not len(sf.e\_entry.get()):

messagebox.showinfo("Missing details","Please fill all the fields to continue.") else:

if sf.p\_entry.get()==sf.p\_entry2.get():

x=sf.cur.execute("select count(\*) from staff where user=%r"%(sf.n\_entry.get())) if list(x)[0][0]!=0:

messagebox.showinfo("Oops !","Username %r already exists.\nPlease try another one."%(sf.n\_entry.get()))

else: try:

sf.cur.execute("insert into staff values(%r,%r,%r,%r)"%(sf.N\_entry.get(),sf.n\_entry.get(),sf.p\_entry2.get(),sf.e\_entry.get())) sf.c.commit()

messagebox.showinfo("Info","You've been successfully registered\nRedirecting to LOGIN window") sf.flush()

sf. init () except: pass

else:

messagebox.showinfo("Mismatched passwords","Both passwords should be same”)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **user** | **passw** | **email** |
| Ruchita  Srivastava | ruchita | 123 | [ruchitasrivastava20@gmail.com](mailto:ruchitasrivastava20@gmail.com) |
| nidhi | nidhi | Nid789 | [nidhi56@gmail.com](mailto:nidhi56@gmail.com) |
| rikitasingh | rikita | Rit78 | [rikitasingh09@gmail.com](mailto:rikitasingh09@gmail.com) |

### TABLE : STAFF

* 1. ***: The Main Page-***

Once the user has logged into the website the main page opens up. Here towards the upper side of the page there are three tabs-

* + - Pizzas
    - Drinks
    - Meals

After logging in, by default the users reach the pizza menu where they can see all the pizzas available along with their description and prices. The users can choose the pizza of their choice by selecting one of the sizes- Regular, Medium, Large. Then by clicking the ‘add to cart’ button the users can add the pizza to the cart. Also, if the users change their mind and want to remove the pizza they selected earlier then they can simply click the remove button to remove it from the cart. The number of times a customer clicks the ‘add to cart’ button decides the quantity of pizzas he/she is ordering. The number etched over the cart figure at the top right corner shows the number of pizzas selected. Also the ‘total’ button beside the cart figure shows the total amount the customers have to pay after selecting the various items they have chosen. The drinks and the meals tab can only be clicked after the customer has chosen some pizza; otherwise if the customer tries to click it without choosing any pizza, the page shows the message asking the customer to select a pizza first. Once the customers have chosen the pizza or pizzas of their choice, if they want they can chose the drink of their choice by clicking the drinks tab which takes them to the menu showing all the drinks available along with their prices.

If they want, the customers can also chose the meals of their choice by clicking the ‘Meals’ tab. This takes them to the meals menu where all the meals available are shown along with their prices and other description.

So after choosing all the items as per their need the customers have to click the check out button at top right corner of the screen and it takes them to the next page.

* 1. ***: The Place Order Page-*** this is the last page that opens up. Here the customers are required to give the address details where they want the pizza to be delivered. The details of whatever items the customers have chosen along with the total amount they have to pay for it is shown to them on this page, so that if they have chosen some item by mistake or want to make any changes they can be reminded so and thus can go back to the previous menus and correct it. After confirming that the order is according to their choice, the customer can click the place order button and the message pops up that their order will be delivered to them at the address they had provided.

### Why Python:

1. **Discussion**

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has the programming easy and fun to do. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features. Python has top the charts in the recent years over other programming languages like C, C++ and Java and is widely used by the programmers. The language has undergone a drastic change since its release 25 years ago as many add-on features are introduced. The Python 1.0 had the module system of Modula-3 and interacted with Amoeba Operating System with varied functioning tools. Python 2.0 introduced in the year 2000 had features of garbage collector and Unicode Support. Python 3.0 introduced in the year 2008 had a constructive design that avoids duplicate modules and constructs. With the added features, now the companies are using Python 3.5.

The software development companies prefer Python language because of its versatile features and fewer programming codes. Nearly 14% of the programmers use it on the operating systems like UNIX, Linux, Windows and Mac OS. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like- Interactive,Interpreted,Modular,Dynamic,Object-oriented,Portable,High level,Extensible in C++ & C.

### Advantages or Benefits of Python

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry. Some of its advantages are-

**Extensive Support Libraries-**

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

## Integration Feature

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other markup languages as it can run on all modern operating systems through same byte code.

## Improved Programmer’s Productivity

The language has extensive support libraries and clean object-oriented designs that increase two to ten fold of programmer’s productivity while using the languages like Java, VB, Perl, C, C++ and C#.

## Productivity

With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications. Python is a robust programming language and provides an easy usage of the code lines, maintenance can be handled in a great way, and debugging can be done easily too. It has gained importance across the globe as computer giant Google has made it one of its official programming languages.

### : Sqlite database-

SQLite is an in-process library that implements a [self-contained](https://www.sqlite.org/selfcontained.html), [serverless](https://www.sqlite.org/serverless.html), [zero-](https://www.sqlite.org/zeroconf.html) [configuration,](https://www.sqlite.org/zeroconf.html) [transactional](https://www.sqlite.org/transactional.html) SQL database engine. The code for SQLite is in the [public](https://www.sqlite.org/copyright.html) [domain](https://www.sqlite.org/copyright.html) and is thus free for use for any purpose, commercial or private. SQLite is the [most](https://www.sqlite.org/mostdeployed.html) [widely deployed](https://www.sqlite.org/mostdeployed.html) database in the world with more applications than we can count, including several [high-profile projects.](https://www.sqlite.org/famous.html)

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database [file format](https://www.sqlite.org/fileformat2.html) is cross-platform - you can freely copy a database between 32- bit and 64-bit systems or between [big-endian](http://en.wikipedia.org/wiki/Endianness) and [little-endian](http://en.wikipedia.org/wiki/Endianness) architectures. These features make

SQLite a popular choice as an [Application File Format](https://www.sqlite.org/appfileformat.html). SQLite database files are a [recommended](https://www.sqlite.org/locrsf.html) [storage format](https://www.sqlite.org/locrsf.html) by the US Library of Congress. Think of SQLite not as a replacement for [Oracle](http://www.oracle.com/database/index.html) but as a replacement for [fopen().](http://man.he.net/man3/fopen)

SQLite is a compact library. With all features enabled, the [library size](https://www.sqlite.org/footprint.html) can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be [faster than](https://www.sqlite.org/fasterthanfs.html) [direct file system I/O.](https://www.sqlite.org/fasterthanfs.html)

SQLite is [very carefully tested](https://www.sqlite.org/testing.html) prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves [100% branch test coverage.](https://www.sqlite.org/testing.html#coverage) SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are [ACID](http://en.wikipedia.org/wiki/ACID) even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides [bugs lists](http://www.sqlite.org/src/rptview?rn=1) and minute-by-minute [chronologies](http://www.sqlite.org/src/timeline) of code changes.

The SQLite code base is supported by an [international team](https://www.sqlite.org/crew.html) of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the [published interface](https://www.sqlite.org/c3ref/intro.html) [spec,](https://www.sqlite.org/c3ref/intro.html) [SQL syntax,](https://www.sqlite.org/lang.html) and database [file format.](https://www.sqlite.org/fileformat2.html) The source code is absolutely free to anybody who wants it, but [professional support](https://www.sqlite.org/prosupport.html) is also available.

The SQLite project was started on [2000-05-09.](https://www.sqlite.org/src/timeline?c=2000-05-29%2B14%3A26%3A00) The future is always hard to predict, but the intent of the developers is to support SQLite through the year 2050. Design decisions are made with that objective in mind.

# :List of Symbols and Abbreviations and Nomenclature

**Def:** in python def is the keyword used to define a function.

**CANVAS:** The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets or frames on a Canvas.

**Def result:** it is a function which through the variable ‘val’ checks which button, whether register or login is clicked and accordingly performs the function related to them.

**Def register:** it is a function that contains code which creates the entire registration page.

**Entry:** The entry widget is used to accept single-line text strings from a user.

**Flush():** Flush function is used to destroy all the previous windows whenever a new window is opened.

**Place:** Place method is used to place the various widgets at their proper place by defining their width, height and the distance from x and y axis.

**Append:** Append method is used for adding the widgets to the window.

### : The Main Concept-

The Pizza Management Website aims at providing the customers an easy interactive environment where they can chose the items of their choice and finally place their order accordingly. The website is quite simple and interactive, the customer thus won’t have any difficulty exploring or navigating through it. It not only provides information about the prices of the items available, but also gives details about the various ingredients each item contains. Along with the pizza the customer can also order the drinks and meals of their choice. They again are available with their prices and description.

The website has been designed using Python and the Sqlite 3 database. In programming, a library is a collection or pre-configured selection of routines, functions, and operations that a program can use. These elements are often referred to as modules, and stored in object format. Libraries are important, because we can load a module and take advantage of everything it offers without explicitly linking to every program that relies on them. They are truly standalone, so we can build our own programs with them and yet they remain separate from other programs. Various libraries have been imported here in making this online pizza ordering system, like SMTPLIB library, Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. The MATH function has been imported to perform operation like addition to calculate the total. Partial function has been imported which can create partial functions in python by using the partial function from the functools library. Partial functions allow one to derive a function with x parameters to a function with fewer parameters and fixed values set for the more limited function. Here it is basically used for handling the mouse events. The Tkinter graphical user interface has been used to create the screens of the application. The three main functions involved in this website are registering,

choosing items of one’s choice and finally placing the order. Registering as already discussed requires the user to provide details like name, username, password and e-mail. After filling which the customer can use the username and password to log into the website. Ordering involves choosing pizzas, drinks and meals by selecting from the products available. Then at last the customers place the order and are confirmed with the message telling that the order will be delivered to them within a short while.

# Future Implications

The online Pizza ordering system for restaurants is a trend which is growing at a fast pace. Online Pizza ordering systems have become a trendsetter, impeccably growing at a faster pace. Today, majority of the restaurants are making use of the online pizza ordering systems that ensure its users to have a better and convenient way of ordering their cuisines. Some of the reasons why online food ordering is becoming popular:

* + - **Bulk orders-** since the online pizza management is simple and very convenient to use website, where you can place the order from anywhere and at any time, thus this flexibility and twenty four hour availability ensures a bulk of customers ordering the pizza online. Unlike the restaurants which can accommodate only a fixed number of customers there is no such constraint and comparatively larger number of customers can order the pizza online. Also unlike the restaurant there is no constraint of time and the customer can order the pizza anytime. These all factors thus result in a bulk of orders.
    - **Improved order accuracy-** while manually taking the orders, sometimes the waiter may commit some mistake and deliver an order which didn’t satisfy the customer’s need. This results in a bad impression of the restaurant, and preparing the order again causes delay for the customer and a lot of wastage of the labour and time of the staff. But that’s not the problem if we are ordering a pizza online because here after the customer has chosen the items they want to be delivered, the place order page shows them or lists out to them the order they have made before they can confirm it. Thus if the order has went wrong somewhere by mistake the customer can correct it before confirming it.
    - **Expanded reach-** various customers who cannot often travel to the restaurant because of long distance problems, or the way from their home to the restaurants involves a lot of traffic, can simply order their pizza sitting at their homes. Thus the online pizza management system allows an extended reach.
    - **Convenient and faster**-as already discuused the online pizza management site is quite flexible and convenient. As soon as an order is placed it gets delivered to the customer under an hour.
    - **Drives more revenue-** as the number of customers are huge it sure results in an increase in the revenue.

The popularity of such pizza ordering websites is estimated to rise in the upcoming years as it directly connects the customers with the restaurant personnel and makes the ordering swiftly. Another positive point about the future of such online pizza website is that there are many enthusiastic business people who want to invest their money somewhere but do not know which trade is the best to put their money that would give them maximum profits. The food industry over the years has grown remarkably and has been growing at an astounding speed. Today every offline industry is following the online system, it is assumed that online food delivery would increase by 30% yearly from 2017 to 2022 across the world. Online food & delivery marketplace has been an encouraging business idea from the start. The Explosive Growth of Online Food Ordering Portals like Food Panda, Just Eat, Grab Hub have made entrepreneurs sit up and take notice. So whether in terms of digitalisation or business or flexibility and convenience or convenience of time and fast delivery, an online pizza system obvious has a great future. Online payment is feature which can also be added to this website so that customer can pay online without involving any cash. Customization of the pizzas and meals is another feature which can also be added to the website. Using this feature the customers would then be able to add the ingredients and modify the pizza as they want. It would be a very valuable addition to the site. Also if robots are involved in taking, managing, processing the orders of such online sites the services can even be faster.

# Conclusion

The goal was to create a website that is user friendly. The website is an easy interface to navigate through and choose the items one wants. The description of all the items as to what the ingredients would be is given in detail along with their price and pictures, so the customers know exactly what they are ordering. Also the size of the pizza can be chosen according to whether the customer wants a regular, medium or large sized pizza. The pizza palace is a python based website. It reduces a lot of time and work of the staff which otherwise has to wasted in taking the orders. The customer can make the order from their place at anytime which reduces them the inconvenience of travelling to some restaurant through the outside traffic and also saves their vehicle’s fuel. Thus ordering the pizza online through this website reduces a lot of staff and customer’s issues thus proves quite economical, flexible and time efficient for both the staff and customer. Many more improvements and future implications can be made to the website as discussed in the future implications part. Thus this website has a wide future scope also.

# Figures:

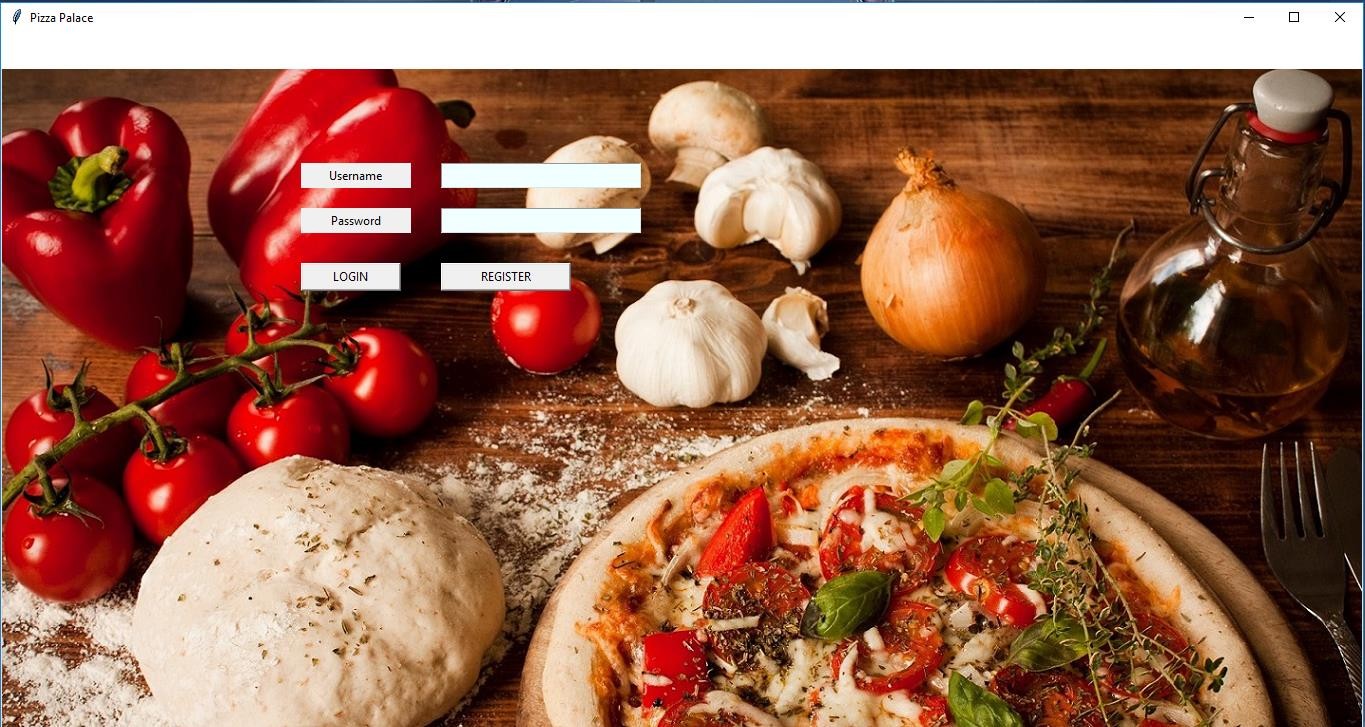


Figure 1: The Opening Page

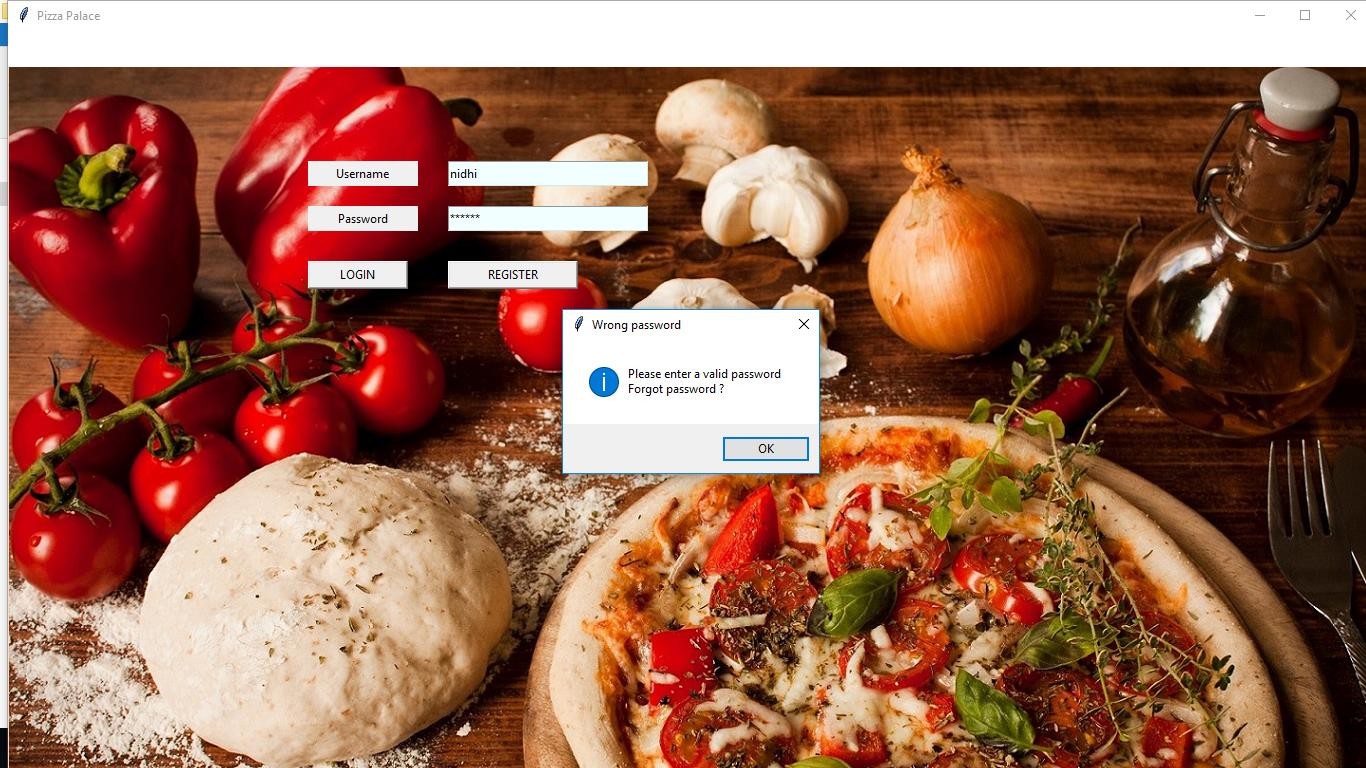


Figure 1.2: opening page-the enter correct password message when you enter some wrong password

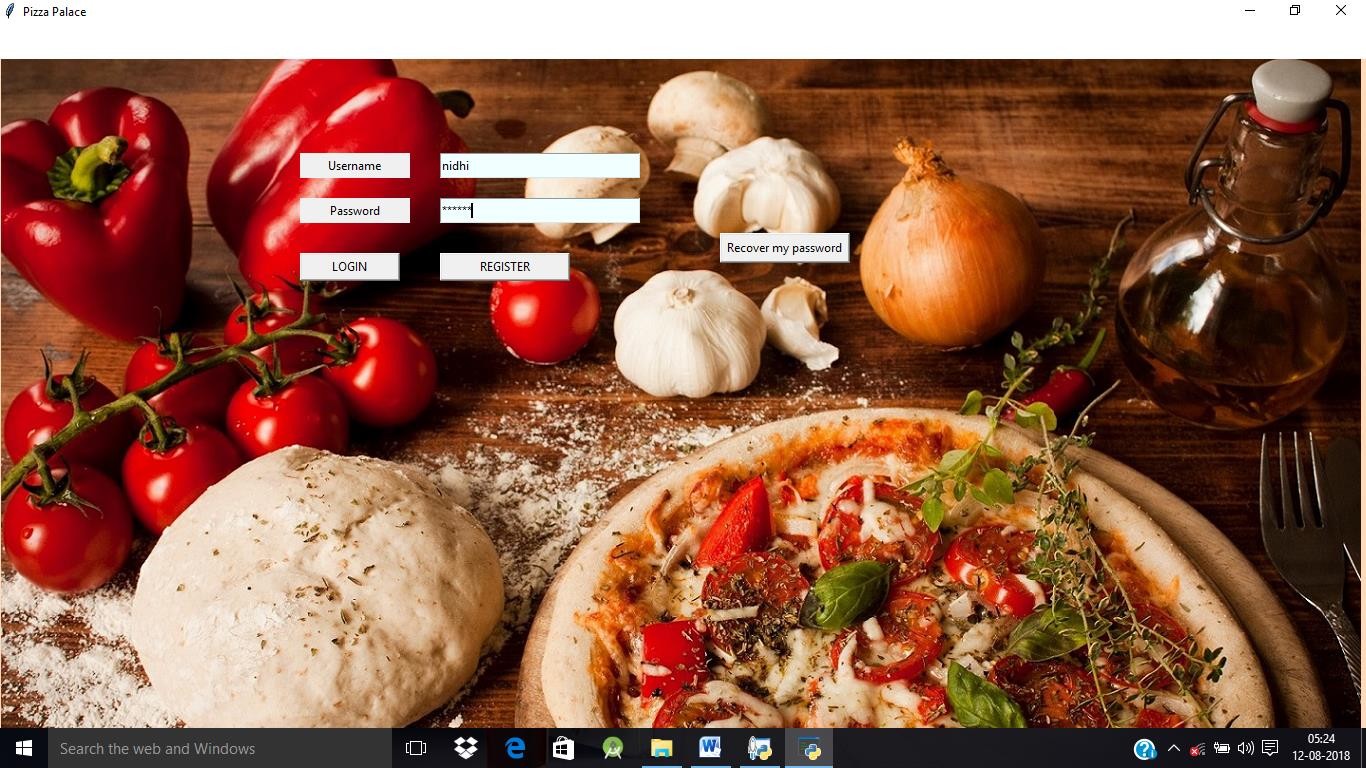


Figure 1.3: opening page-the recover password button helps to recover customer’s password in case they forget it.

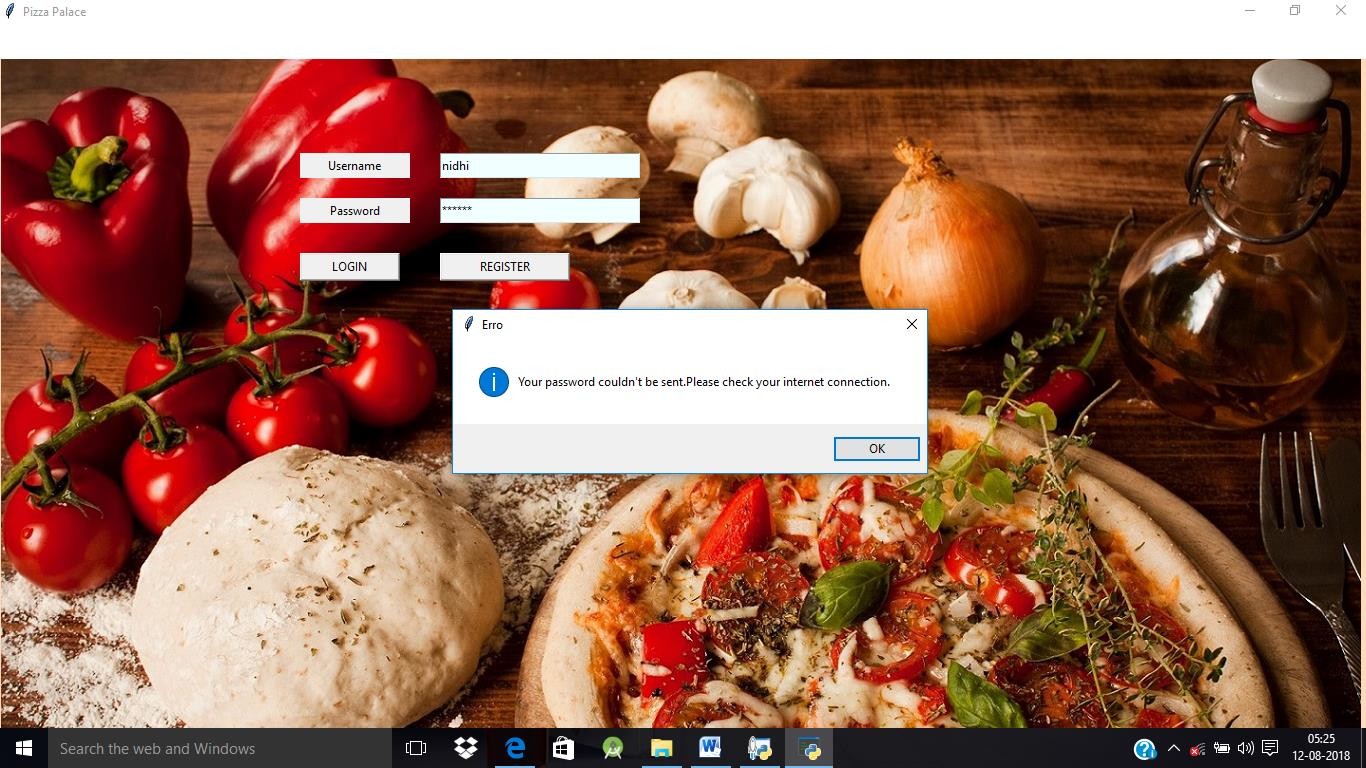


Figure 1.4: if the new password couldn’t be send then it shows the notification specifying the same and asks the customers to check their internet connection.

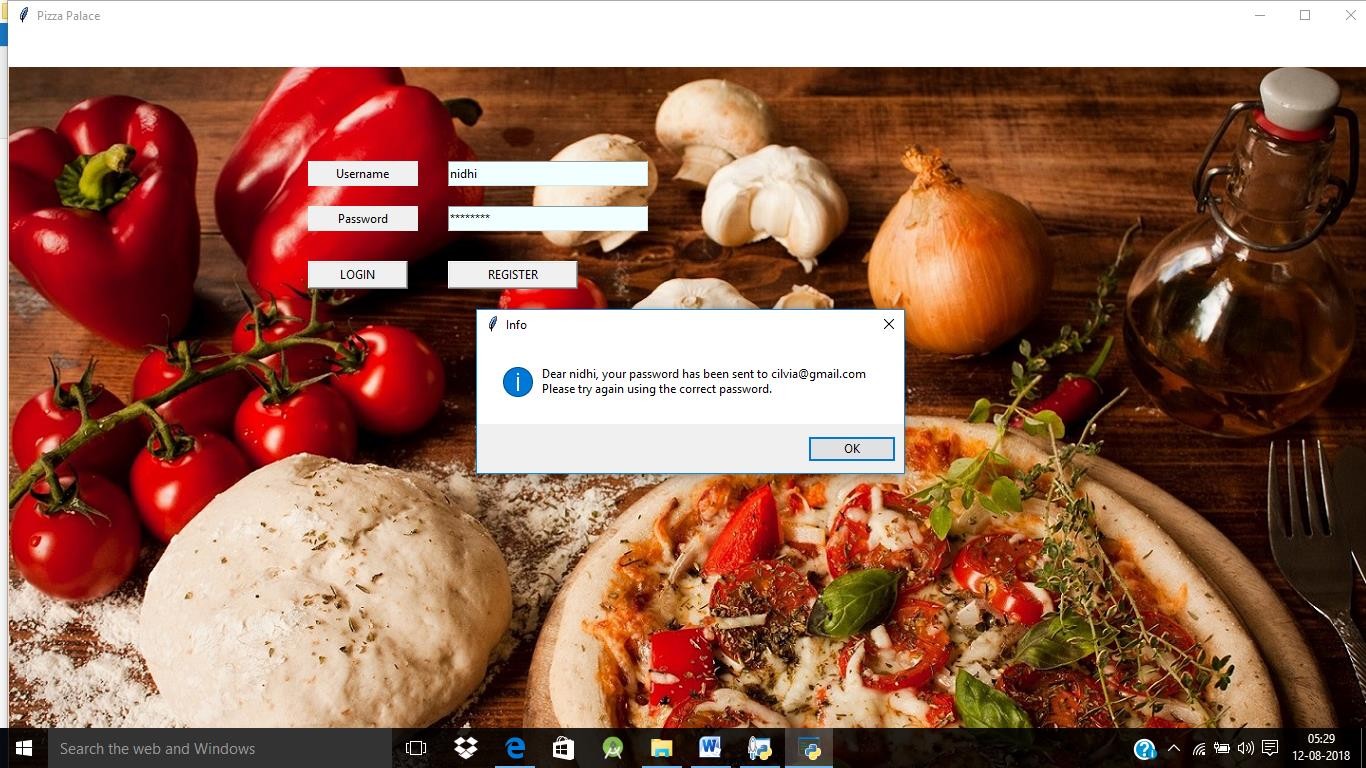


Figure 1.5: If the password is sent successfully then the notification shows up for the same.

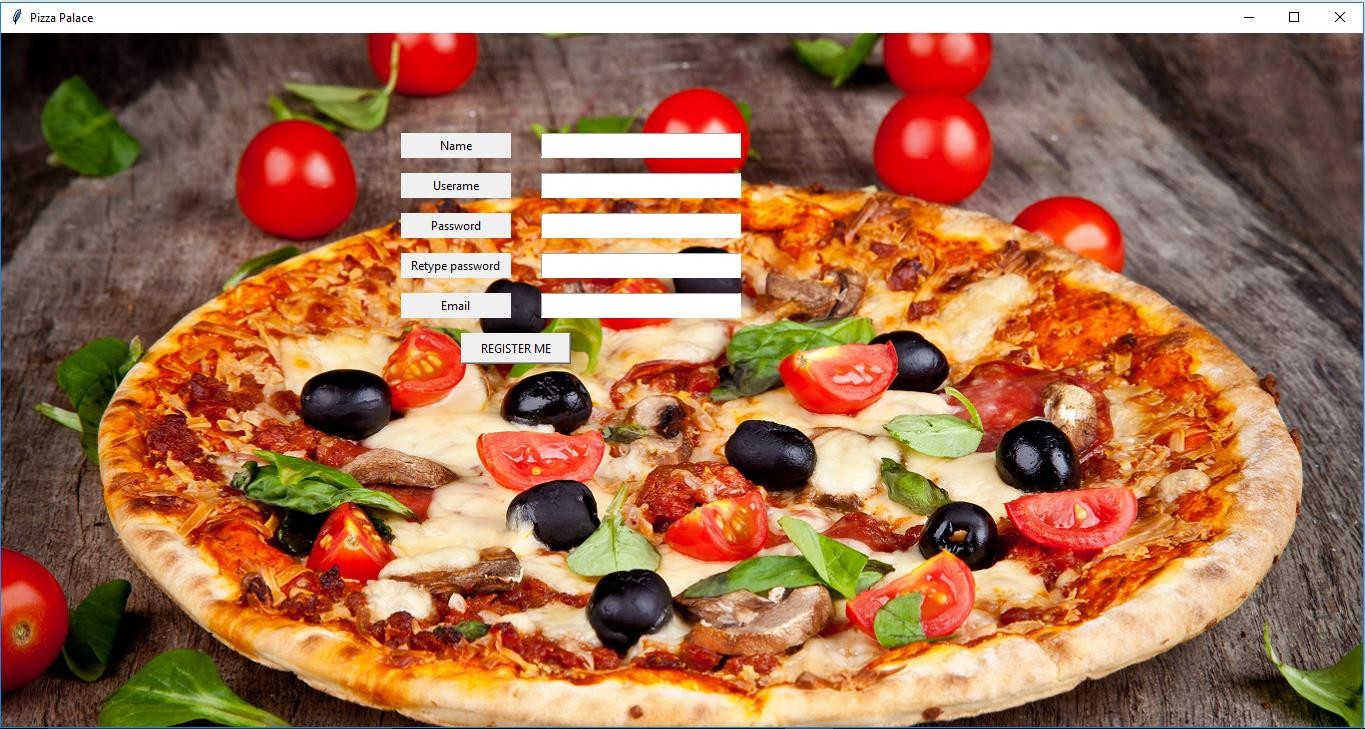


Figure 1.6: The Registration Page

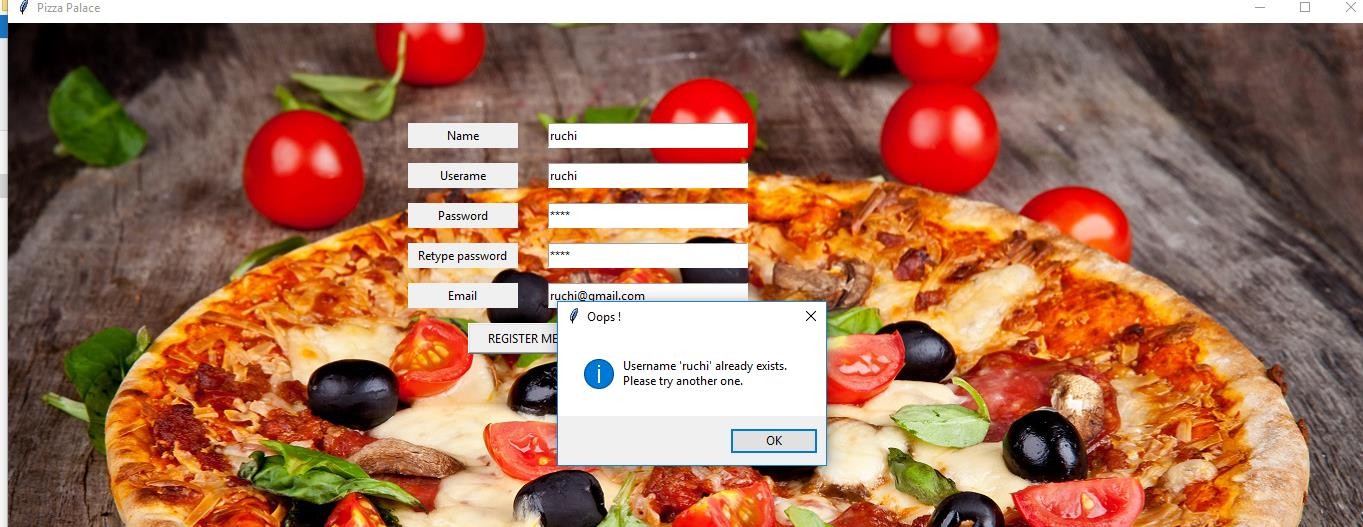


Figure 2.2: If the user enters an existing username, the application asks to enter another username.

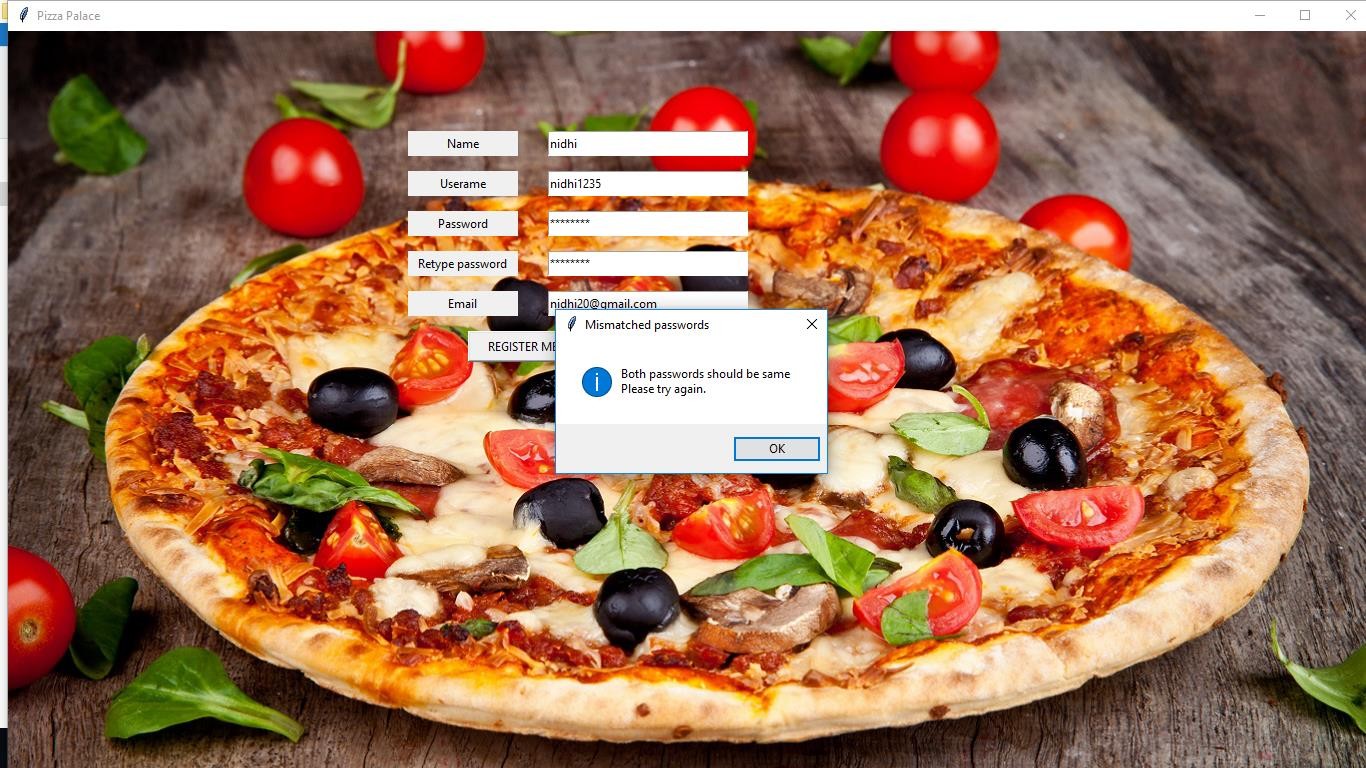


Figure 2.3: If the password and retype password do not match, the application asks to make these two entries same.

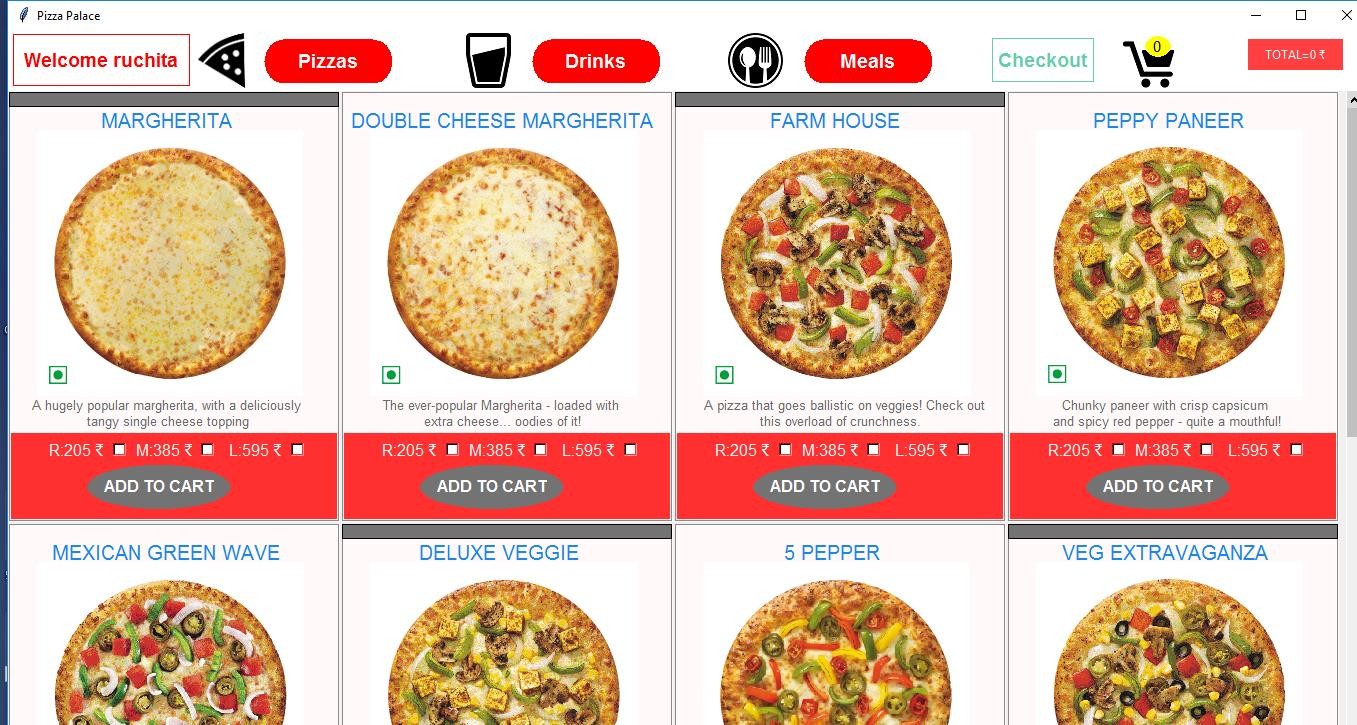


Figure 3.1: The Main Page-Pizza Menu

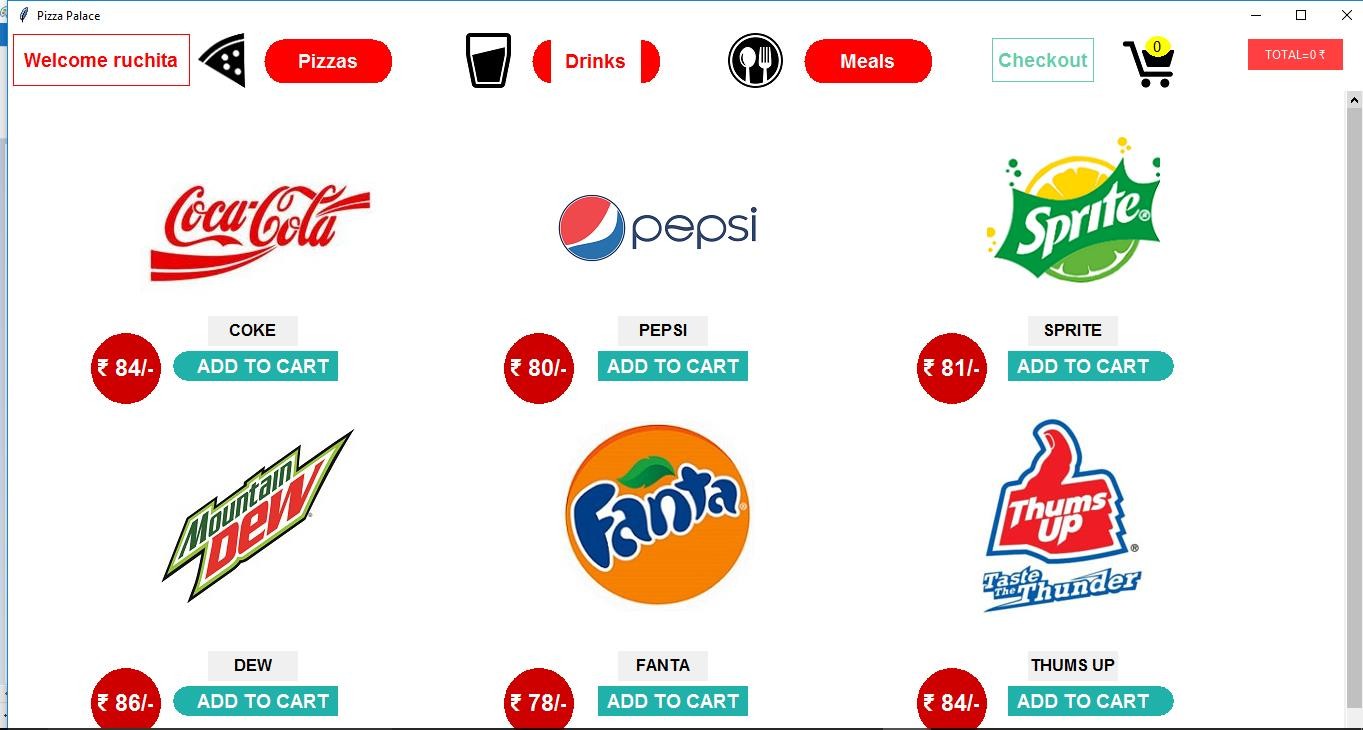


Figure 3.2: Main Page-Drinks Menu

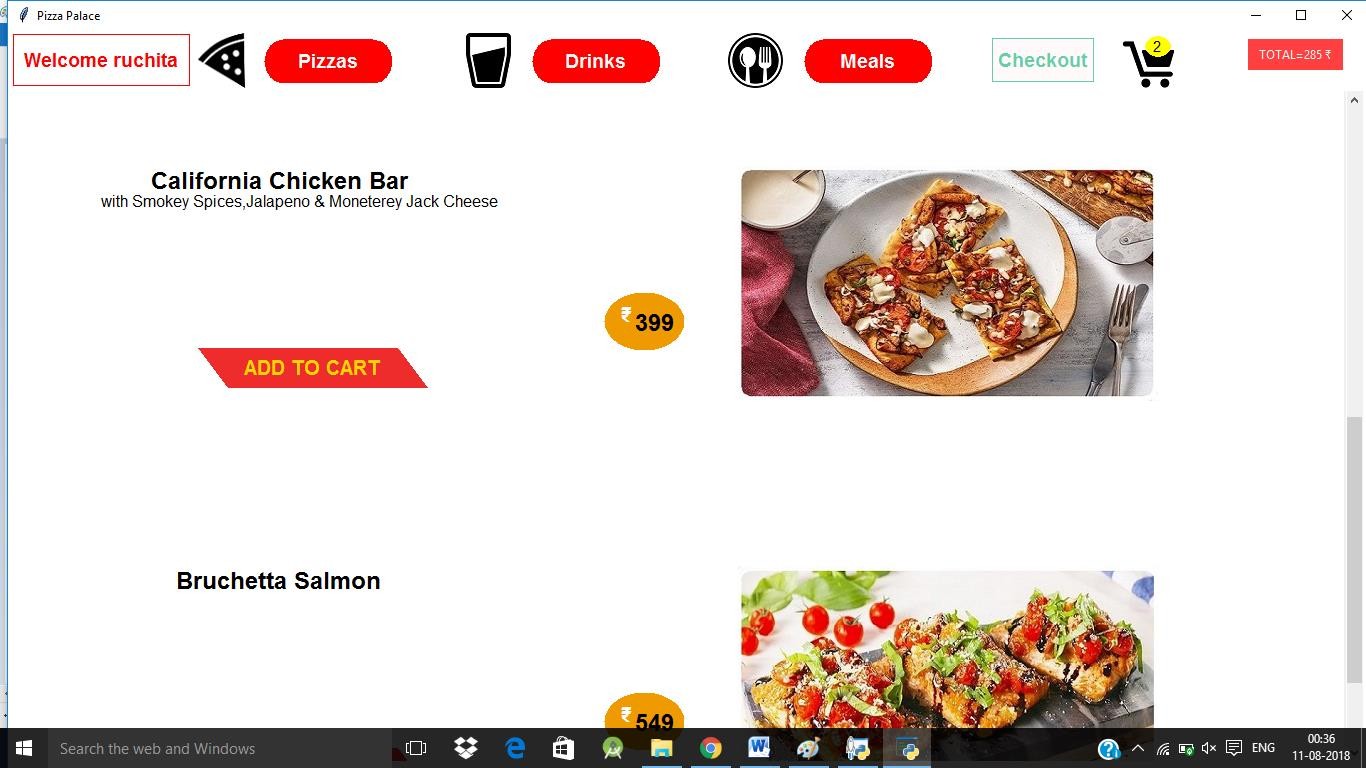


Figure 3.3: Main Page-Meals Menu

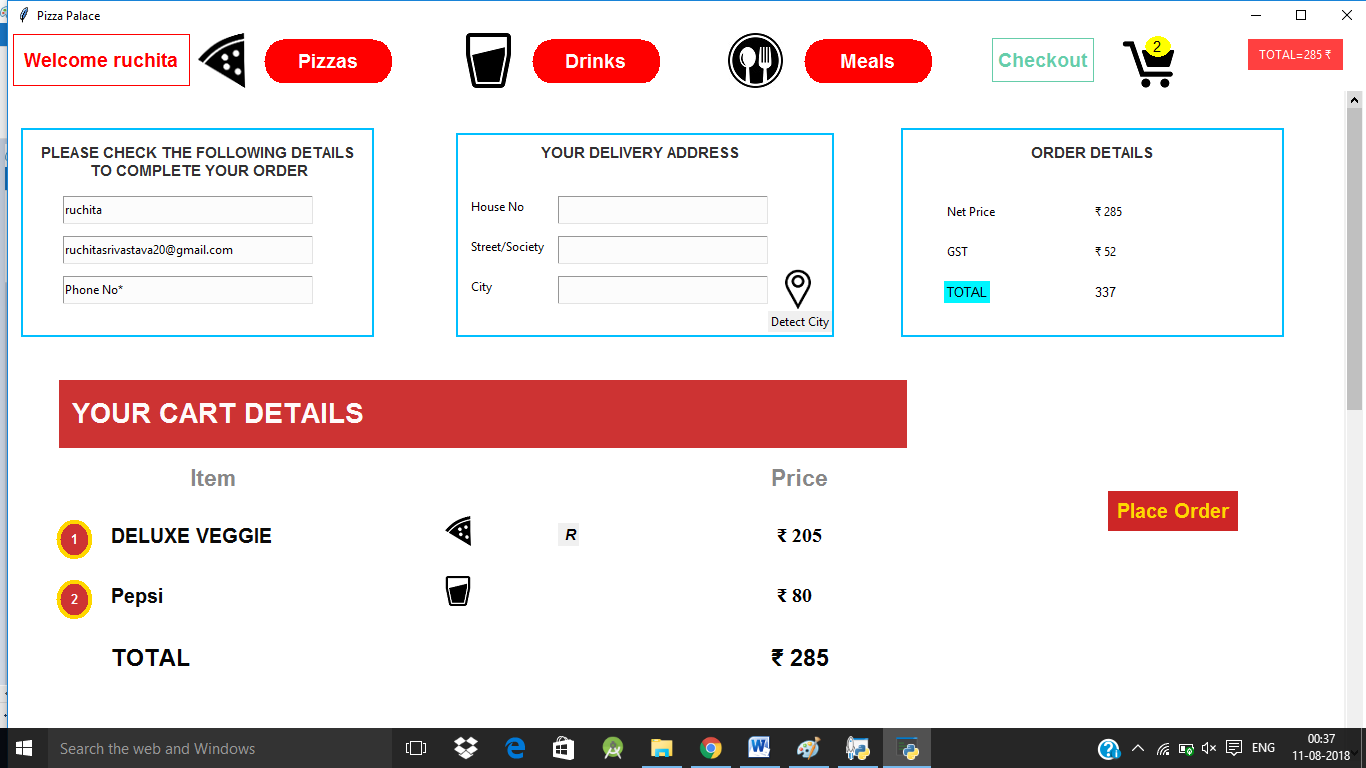


Figure 4.1: The Place Order Page

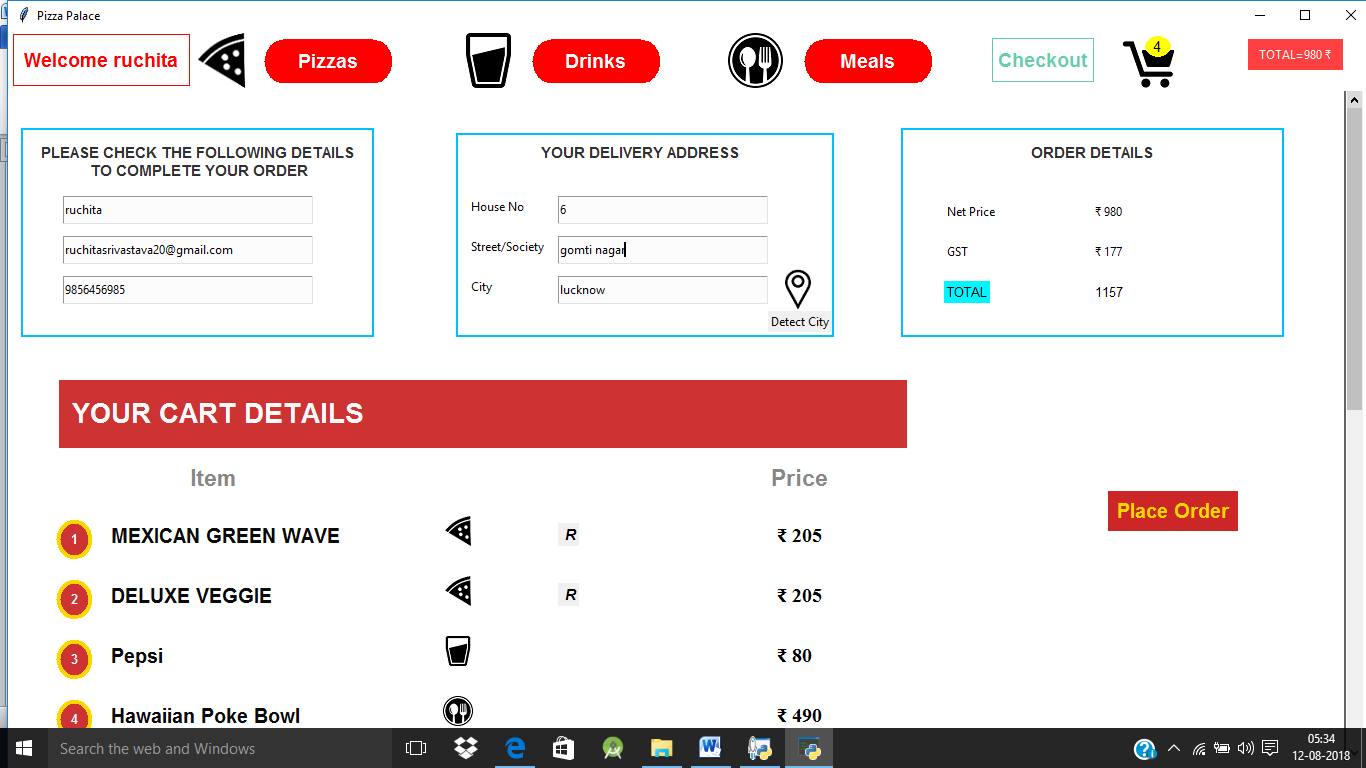


Figure 4.2: Place Order Page-In delivery address the customer needs to enter all the details like House No., Street and City.

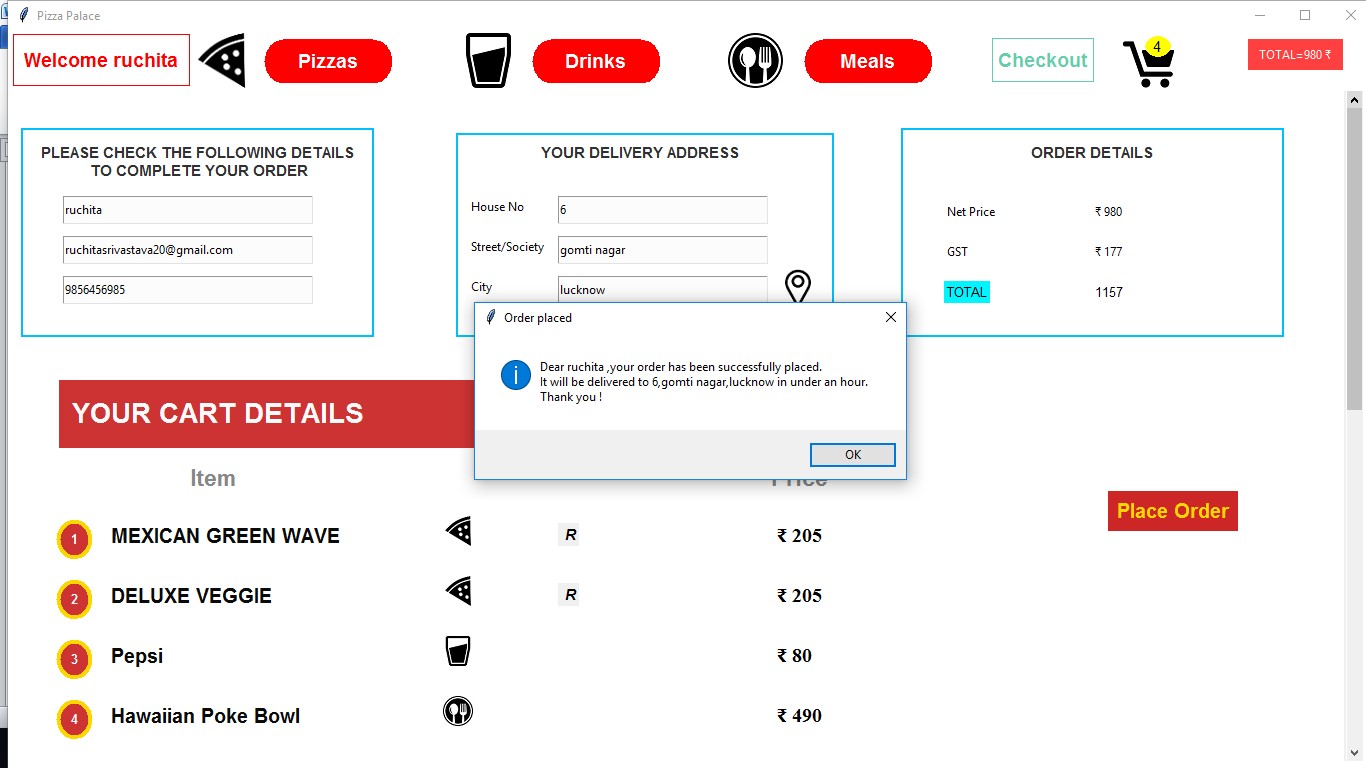


Figure 4.3: once the customer has clicked the place order button the order gets placed and the notification that it will be delivered to customer in an hour shows up.

# References

* + - * https://en.wikipedia.org//python
      * https://en.wikipedia.org
      * htt[ps://www.lovel](http://www.lovelycoding.org/2015/11/online-pizza-ordering-system)y[coding.org/2015/11/online-pizza-ordering-system](http://www.lovelycoding.org/2015/11/online-pizza-ordering-system)
      * [www.scribd.com](http://www.scribd.com/)
      * [https://stackoverflow.com](https://stackoverflow.com/)
* htt[ps://www.p](http://www.python.org/)y[thon.org](http://www.python.org/)