**WEB APPLICATION**

**FOR**

# CANTEEN AUTOMATION

**By**

|  |  |
| --- | --- |
| **BHINGARADIYA VIVEK** | **14CEUOG025** |
| **GAUSWAMI DARSHAN** | **14CEUBS105** |

**A project submitted for**

**PILOT PROJECT**

**In**

**Computer Engineering**

**TECHNOLOGY: PYTHON**

**DATABASE: MYSQL**

**FRAMEWORK: DJANGO**

**Under The Guidance of**

Prof. Brijesh S. Bhatt



**DEPARTMENT OF COMPUTER ENGINEERING,**

**FACULTY OF TECHNOLOGY,**

**DHARMSINH DESAI UNIVERSITY**

**NADIAD-387001, GUJARAT**

**APRIL 2018**

**DHARMSINH DESAI UNIVERSITY**

**NADIAD-387001, GUJARAT**



CERTIFICATE

This is to certify that the project carried out in the subject of PILOT Project entitled “**CANTEEN AUTOMATION”** and recorded in this report is a work of

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **(1)** | **BHINGARADIYA VIVEK** | **14CEUOG025** | **CE08** |
| **(2)** | **GAUSWAMI DARSHAN** | **14CEUBS105** | **CE35** |

Carried out in Pilot Project in Computer Engineering at Dharmsinh Desai University in the academic session December 2017 to April 2018.

|  |  |
| --- | --- |
| Prof. Brijesh S. Bhatt | Dr. C.K.Bhensdadia, |
| Department of Computer Engineering, | Head, Department of Computer Engineering, |
| Faculty of Technology, | Faculty of Technology, |
| Dharmsinh Desai University, Nadiad | Dharmsinh Desai University, Nadiad |

**Acknowledgement**

It is our privilege to express our sincerest regards to our project coordinator, **Prof. Brijesh S. Bhatt** for their valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. We deeply express our sincere thanks to our Head of Computer Department **Dr. C.K.Bhensdadia** for encouraging and allowing us to present the project on the topic “**CANTEEN AUTOMATION**“at our department premises for **PILOT PROJECT** in **Computer Engineering.** Last but not the least we express our thanks to all our lecturers and friends who have directly or indirectly helped our project.

Thanks and Regards,

Vivek Bhingaradiya &

Darshan Gauswami

**ABSTRACT**

In India and other countries of world, because of the management of canteen not automated. So to solve this problem CANTEEN AUTOMATION is useful.

For this application there are three types of users’Admin (manager), chef and customers. In this application people (customers) have to place order online based on category of items, they have to choose required quantity and pay total amount online. Chef will get all orders based on category with respect to chef category. Chef update status of order as per preparation of order. Manager manages chefs, items, orders and order details.

Table of Contents

[CANTEEN AUTOMATION 1](#_Toc504434234)

[ACKNOWLEDGEMENT 3](#_Toc504434235)

[ABSTRACT 4](#_Toc504434236)

[Table of Contents 5](#_Toc504434237)

[1. INTRODUCTION 6](#_Toc504434238)

[1.1. Purpose 6](#_Toc504434239)

[1.2. Document Convention 6](#_Toc504434240)

[1.3. Intended Audience and Reading Suggestions 6](#_Toc504434241)

[1.4. Product Scope 6](#_Toc504434242)

[1.5. References 6](#_Toc504434243)

[2. ABOUT THE SYSTEM 7](#_Toc504434244)

[2.1. Modules 7](#_Toc504434245)

[2.2. Product Functions 7](#_Toc504434246)

[2.3. Requirements 7](#_Toc504434247)

[2.4. Operating Environment 10](#_Toc504434248)

[3. ANALYSIS & DESIGN 11](#_Toc504434249)

[3.1. ER Diagram 11](#_Toc504434250)

[3.2. Use Case Diagram 12](#_Toc504434251)

[3.3. Class Diagram 13](#_Toc504434252)

[3.4. Activity Diagram 14](#_Toc504434253)

[3.5. Sequence Diagram 16](#_Toc504434254)

[3.6. State Diagram 17](#_Toc504434255)

[3.7. Activity Network Diagram 19](#_Toc504434256)

[4. SQL DDL & DATA DICTONARY 20](#_Toc504434257)

[4.1. SQL DDL 20](#_Toc504434258)

[4.2. Data Dictionary 22](#_Toc504434259)

[5. IMPLEMENTATION 25](#_Toc504434260)

[5.1. Installation of Python 25](#_Toc504434261)

[5.2. Installation of Django 26](#_Toc504434262)

[5.3. Python Command Cell Access 27](#_Toc504434263)

[5.4. Admin 28](#_Toc504434264)

[5.5. Client 29](#_Toc504434271)

[5.6. Chef 30](#_Toc504434277)

[5.7. Payment 31](#_Toc504434280)

[5.8. Exception Handling 31](#_Toc504434281)

[5.9. Logout 31](#_Toc504434282)

[6. TEST CASE DESIGN 32](#_Toc504434283)

[6.1. Testing Plan: 32](#_Toc504434284)

[6.2. Testing Strategy: 32](#_Toc504434285)

[6.3. Testing Methods 32](#_Toc504434286)

[6.4. Test Cases: 33](#_Toc504434287)

[7. SCREENSHOTS 37](#_Toc504434288)

7.1. [Customer Login & Registration 37](#_Toc504434289)

7.2. [Forget Password & Choose Order 38](#_Toc504434290)

7.3. [Add To Cart & Place Order 39](#_Toc504434291)

7.4. [Confirm Order& Paypal Payment 40](#_Toc504434292)

7.5. [Cancel Payment 40](#_Toc504434293)

7.6. [Add & View Item 42](#_Toc504434294)

7.7. [Update Item & Add Chef 43](#_Toc504434295)

7.8. [View Chef & View Order 44](#_Toc504434296)

7.9. [View Orderdetails & View Cutomer 45](#_Toc504434297)

7.10. [Manage Order & Prepare History 46](#_Toc504434298)

[8. CONCLUTION AND FUTURE EXTENTION 47](#_Toc504434299)

[9. BIBLOGRAPHY 48](#_Toc504434300)

CHAPTER

# INTRODUCTION

## Purpose

The purpose of our canteen automation is to provide facilities to order online food & online payment. The scope of our project is that it can be provide food without standing in queue.

## Document Convention

IEEE standard for SRS.

## Intended Audience and Reading Suggestions

System has different types of reader that the document is intended for such as developer, project manager, testers and documentation writer. The SRS has been made in order of increasing specificity. Using this SRS customers and their staff can easily understand the features of software and can use it efficiently for more productive.

## Product Scope

* Easy to use
* Increase the speed of management of order
* Online payment using paypal account only.
* User friendly interface
* User can see all the history of orders placed by him/her.

## References

IEEE standard for SRS and software engineering chapter 5.

CHAPTER

1. ABOUT THE SYSTEM

## Modules

* There are three different modules are in the system.
* Admin(Manager)
* Chef
* Customer
* Payment

## Product Functions

* **Admin Panel**
* Admin can manage items, chefs, orders and orderdetails. Admin can view all customers as well as update and delete items and chefs.
* **Chef’s panel**
* All chefs can be registered by admin only. There must be one category per one chef. After login within any department he/she can prepare order within his/her category. Any chef cannot manage order of other category apart from his/her category. Chef can update status based on status of preparing order.
* **Customer’s panel**
* All customers can register themselves. They can see all items after login successfully. Customers can see the all items differentiate based on category. They have to select items and quantity. Customers can view history of their placed order. Customer can cancel order if status of order is pending.

## Requirements

**Registration and Login page:**

* + 1. **Description and Priority**

Registration is for the users who want to be associated with the web-app for long period to avail long term benefits.

Login page is for already registered users.

* + 1. **Stimulus/Response Sequences**

System provides two option on the first page – Register and Login

If the user selects register then the registration page will be displayed where user has to fill all the details. After successful registration new account will be created for that user.

If the user selects Login option the user will be asked for user name and password and will be directed to his/her personalized home page.

* + 1. **Functional Requirements**

REQ-1: System should provide registration page.

Input: User Details

Output: Registration Status

Process: System takes all the user details, does the input validation and adds to the data database hence new user account is created.

Error Condition: If user details are incomplete or in improper format the System redirects to the same page and asks user to refill data.

REQ-2: System provides Login Page

Input: User name and password

Output: Validation Status

Process: System takes user name, password and verifies with the database.

Error Condition: If user name or password is incorrect then system gives appropriate message.

REQ-3: System should provide reset password facility.

Input: Username or email id, password and new password and confirm password.

Output: password status

Process: System takes username or email and password . then it will take to change password page then system should check weather new password and confirm password are same or not. If same then system should update the database.

Error Condition: If user enters incorrect password or not in required form then should give appropriate message.

**Form for choose order for customer and manage order for chef forms**

REQ-4: System must have categories where the items and orders are selected and managed.

Input: Select category, item and quantity.

Output: Status of the order.

Process: Enter the details of the spent money into database and give output.

Error Condition: If the internet connection lost then error occur.

REQ-5: System must have categories for which chef can manage order.

Input: Chef Id and chef name.

Output: Order list with status.

Process: Validate the chef and check category for him/her. View list of

Order with status.

Error Condition: If the internet connection lost then error occur.

**Forms for Admin**

REQ-6: System should provide Add item and Add chef for admin.

Input: Item details for add item and Chef Details for add chef

Output: add item of chef Status

Process: System takes all the item or chef details, does the input validation and adds to the data database hence new item or chef is created.

Error Condition: If Item or Chef Details are incomplete or improper format the System redirects to the same page and asks Admin to refill data.

REQ-7: System should provide views for item, chef, customer, order and order details.

Input: Select appropriate view navigator.

Output: Details in form of table.

Process: Fetch the details from the database and show into the form of table.

## Operating Environment

Software specification of the system that used for this project is,

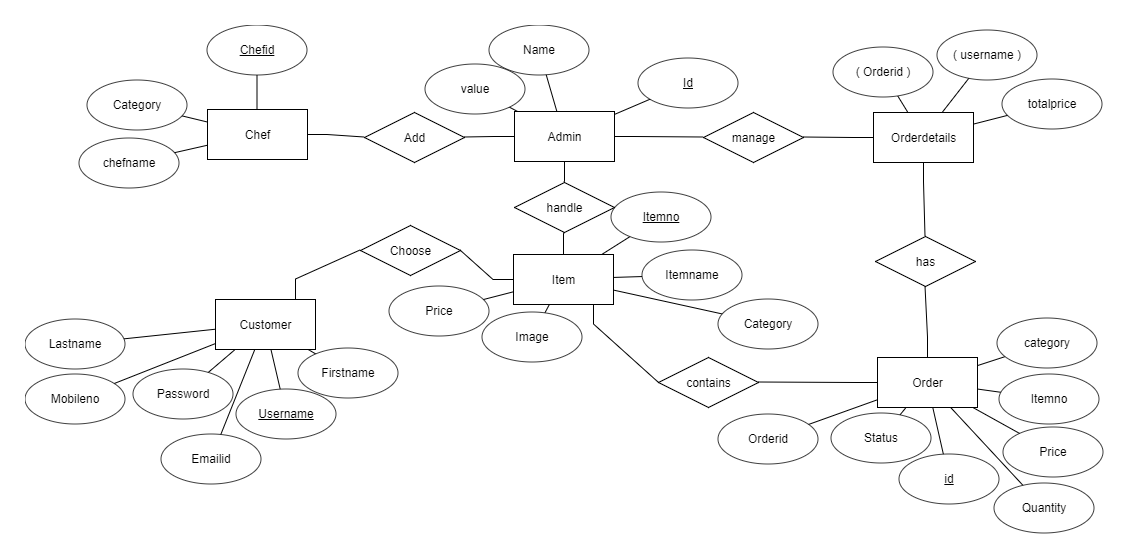
Operating System : Any OS on which able to install python and Django and apache for xampp or wampp.

For Database : MySQL Database

CHAPTER

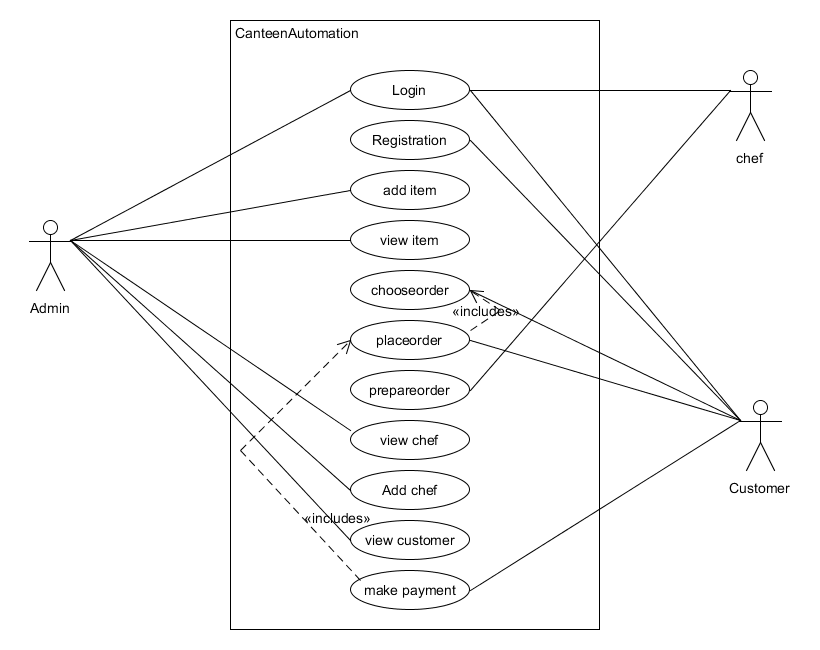
1. ANALYSIS & DESIGN

## ER Diagram

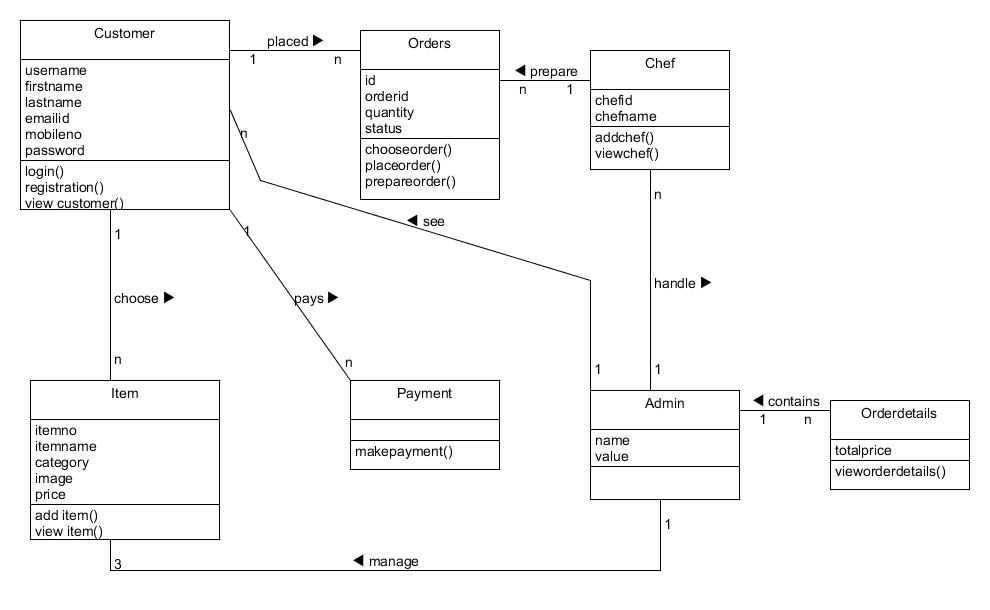
****

<https://erdplus.com/#/edit-diagram/468281>

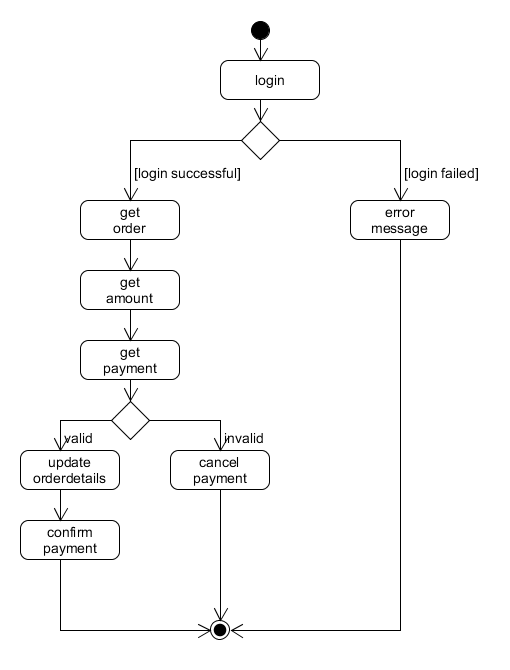
## Use Case Diagram

****

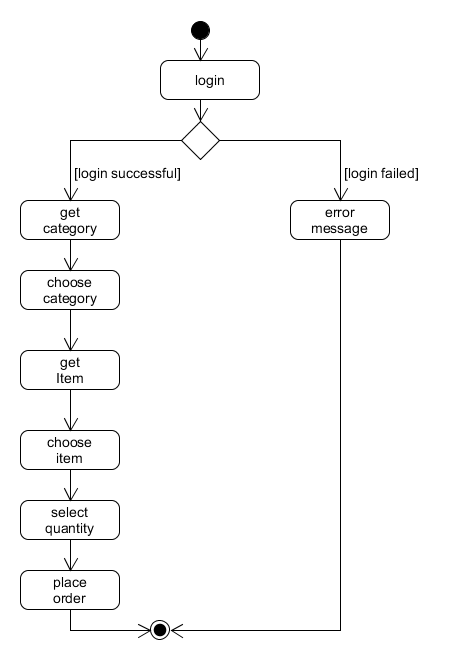
## Class Diagram

****

## Activity Diagram

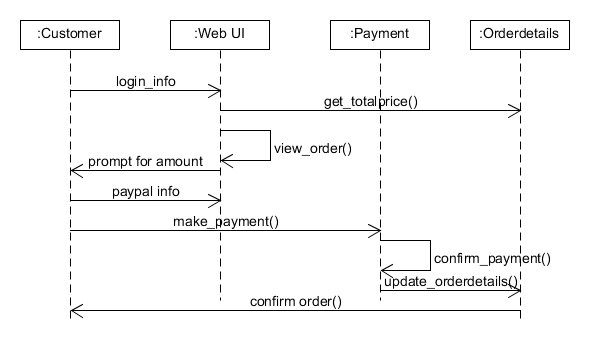
****

Payment Activity Diagram

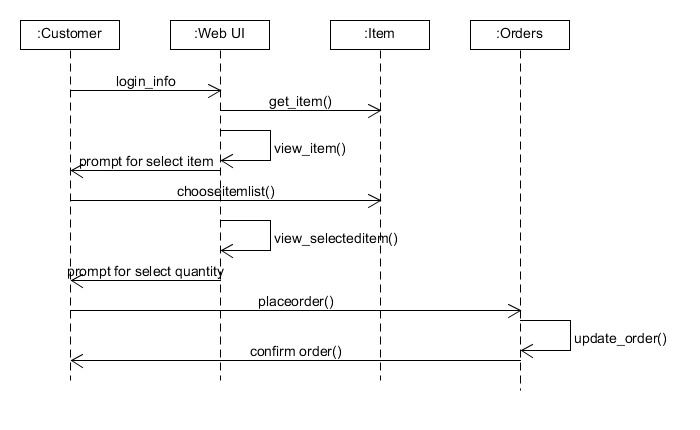


Place Order Activity Diagram

## Sequence Diagram

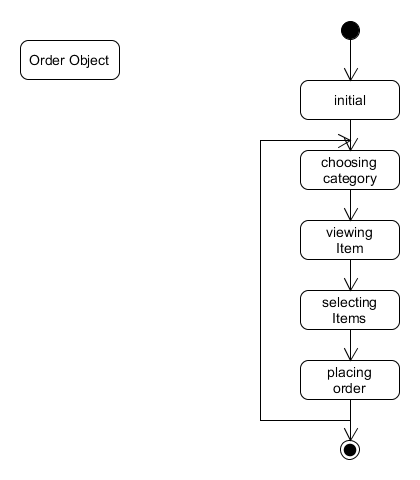


Payment Sequence Diagram

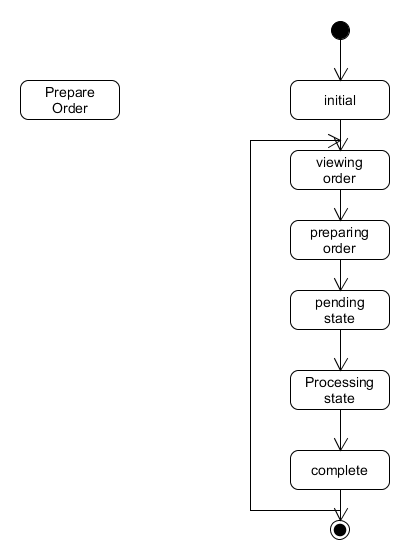


Place Order Sequence Diagram

## State Diagram



State Diagram for Placeorder



State Diagram for Prepare Order

## Activity Network Diagram

SPECIFICATION 1

DESIGN DATABASE 3

DESIGN UI 3

WRITE MANUAL 38

CODE GUI 27

CODE DATABASE 25

INTEGRATE & TEST 35

FINISH 39

CHAPTER

1. SQL DDL & DATA DICTONARY

## SQL DDL

* + 1. **Admin**

CREATE TABLE `admin` (

`id` int(11) NOT NULL,

`name` varchar(10) NOT NULL DEFAULT 'admin',

`value` varchar(10) NOT NULL DEFAULT 'admin'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + 1. **Chef**

CREATE TABLE `chef` (

`chef\_id` varchar(10) NOT NULL,

`password` varchar(10) NOT NULL,

`category` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + 1. **Item**

CREATE TABLE `item` (

`itemno` int(11) NOT NULL,

`itemname` varchar(20) NOT NULL,

`category` varchar(20) NOT NULL,

`price` int(11) NOT NULL,

`image` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + 1. **Orders**

CREATE TABLE `orders` (

`orderid` int(11) NOT NULL,

`itemno` int(11) NOT NULL,

`category` varchar(15) NOT NULL,

`quantity` int(11) NOT NULL,

`price` int(11) NOT NULL,

`status` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + 1. **Orderdetails**

CREATE TABLE `orderdetails` (

`orderid` int(11) NOT NULL,

`username` varchar(20) NOT NULL,

`totalprice` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + 1. **Customers**

CREATE TABLE `customer` (

`username` varchar(10) NOT NULL,

`firstname` varchar(20) NOT NULL,

`lastname` varchar(20) NOT NULL,

`password` varchar(10) NOT NULL,

`mobileno` varchar(10) NOT NULL,

`emailid` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**Django Models**

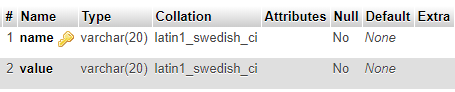
**class** Item(models.Model):  
 itemno = models.CharField(max\_length=20,primary\_key=**True**)  
 itemname = models.CharField(max\_length=20)  
 category = models.CharField(max\_length=20)  
 price=models.IntegerField  
 image=models.CharField(max\_length=200)  
  
**class** Customer(models.Model):  
 username = models.CharField(max\_length=20,primary\_key=**True**)  
 firstname = models.CharField(max\_length=20)  
 lastname = models.CharField(max\_length=20)  
 password=models.CharField(max\_length=20)  
 mobileno=models.CharField(max\_length=10)  
 emailid=models.EmailField(max\_length=30)  
  
**class** Orders(models.Model):  
 orderid = models.CharField(max\_length=20,primary\_key=**True**)  
 itemno = models.CharField(max\_length=20)  
 category = models.CharField(max\_length=20)  
 quantity=models.IntegerField  
 price = models.IntegerField  
 status = models.CharField(max\_length=10)

**Class** Admin (models.Model):  
 name = models.CharField (max\_length=20)  
 value = models.CharField (max\_length=20)  
  
**class** Orderdetails(models.Model):  
 orderid = models.CharField(max\_length=20)  
 username = models.CharField(max\_length=20)  
 totalprice=models.IntegerField

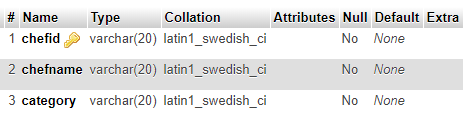
**Class** Chef (models.Model):  
 chefid = models.CharField (max\_length=20, primary\_key=**True**)  
 chefname = models.CharField (max\_length=20)  
 category = models.CharField (max\_length=20)

## Data Dictionary

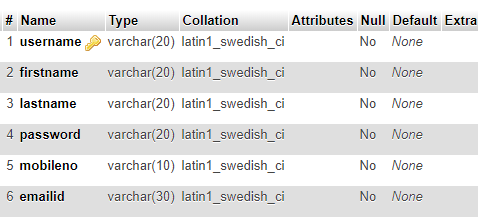
* + 1. **Admin**



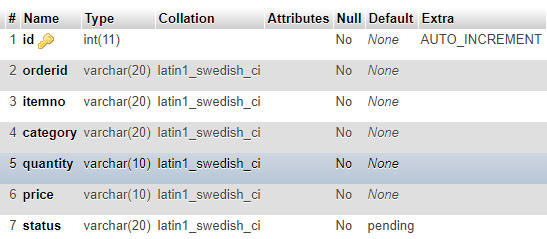
* + 1. **Chef**



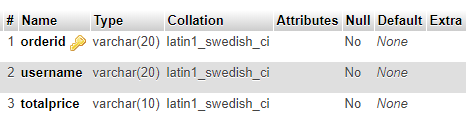
* + 1. **Customer**



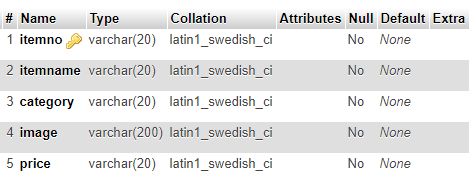
* + 1. **Orders**



* + 1. **Orderdetails**

****

* + 1. **Item**



CHAPTER

# IMPLEMENTATION

We are implementing this system using python technology and Django framework. We are using MYSQL database with it. All the frontend design is done is in html using either bootstrap or materialize css and JavaScript.

For the given project, we are using following versions

Python - 3.6.3

Django - 2.0.0

**5.1 Installation of Python**

For Windows users:

You can download required .exe file from this link and then execute downloaded file for installation.

For MAC OS users:

For Mac OS X 10.6 and later they can download required installer from this link and install that package.

For UNIX / Linux users:

For users who are using newer version of Linux / UNIX they can use following command:

$ sudo apt-get update

$ sudo apt-get install python3.6

If anyone who are not able to install by above command, they can use below command:

$ sudo apt-get install software-properties-common

$ sudo add-apt-repository ppa: deadsnakes/ppa

$ sudo apt-get update

66$ sudo apt-get install python3.6

**Installation of Pip**

Pip is a package manage for Python. It makes installing and uninstalling Python packages (such as Django) very easy. To install a pip use following command.

$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

$ python get-pip.py

**5.2 Installation of Django**

1. Before installing Django, verify that python is already installed into your computer and PATH variable is set.

Check if python is installed by using the following command in command prompt.

-python -V

It should return the version of installed python. If not, install python and check again.

2. Install Django

Django can be installed easily using pip within your virtual environment.

In the command prompt, ensure your virtual environment is active, and execute the following

* **command:**

**-pip install django**

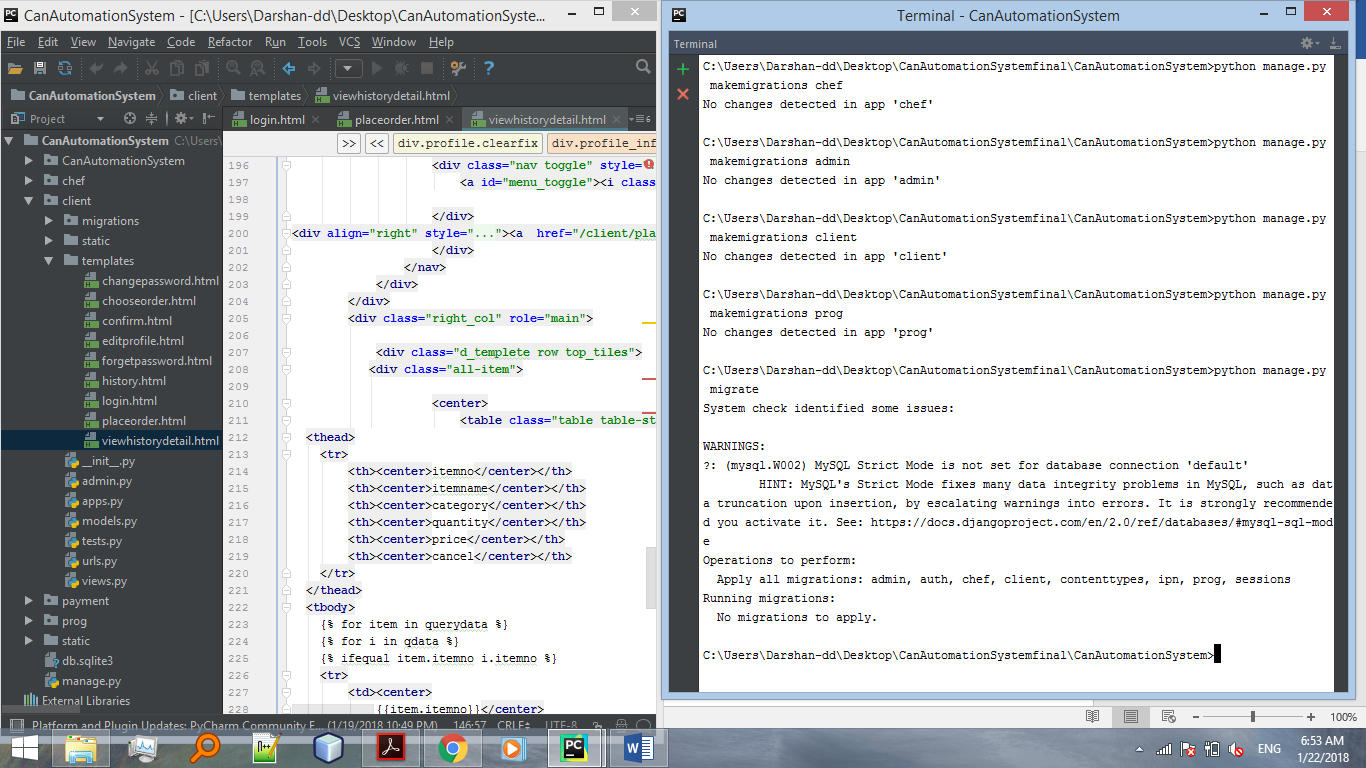
This will download and install the latest Django release.

We can verify it by using django-admin --version in the command prompt.

We can get errors in case if the PATH system environment variable is not set to python directory.

**Command for migration of models to database: -**

* **python manage.py makemigrations appname**
* **python manage.py migrate**



**5.3 Python Command Cell Access**

There are two ways for access python command cell

**1.** **Command prompt**

**2.** **Pycharm terminal**

**1. Command Prompt**

* -open command prompt
* -set path to your python installed directory

**2. Pycharm terminal**

* -Install and open Pycharm
* -go to view menu
* -select Tool Windows
* -select terminal
* **New Project creation**

**Command:**

**-django-admin.py startproject projectname**

**OR**

**-django-admin startproject projectname**

* **New app creation in project**

Go to project directory

**Command:**

-**python manage.py startapp appname**

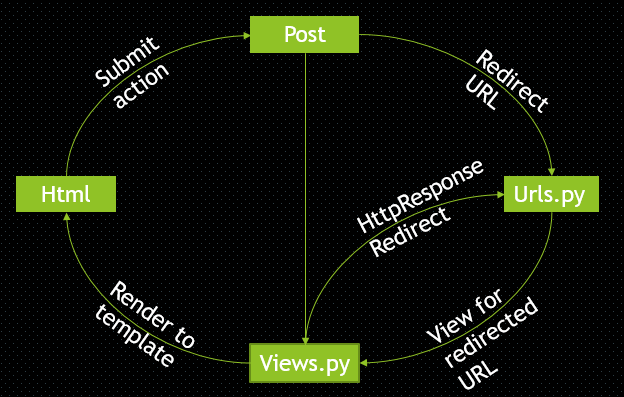
* **Run Project**

**Command:**

-**python manage.py runserver**

**server ip address**

**127.0.0.1:8000**

****

**TRAVARSAL IN DJANGO**

## 5.4 Admin

Admin has access to add, delete or update item, add or remove chef and all the details of customer who has ordered.

* + 1. **Add Item**

Admin can add the details about items i.e. its id, name, category, images and price. Added item will be shown to user. All the details are validated in jQuery.

Error will shown as alert at same page. If all the details are proper then it will show item successfully added. Unique itemno will be checked dynamically using jQuery.

### **5.4.2 View Item**

Here Admin can view all item which is added previously. Admin can update or delete the item as well from here. When admin clicks on update button it will redirect to update item template. Where it show all previous details of Item. Admin will update these previous values and press Update button then all details are validated through jQuery if any error occure then it will show it in alert box and then redirect to same page. If all details are proper then it will show Item updated successfully. When Admin clicks on delete button it will delete that item from database and again redirect to view item page.

* + 1. **Add Chef**

Admin can add the details about chef i.e. its id, name and category. All the details are validated in jQuery.Error will shown as alert at same page. If all the details are proper then it will show item successfully added.Unique itemno will be checked dynamically using jQuery.

### **5.4.4 View Chef**

Here admin can view all chef and its details which is currently working as well as admin has access to update the details of chef and can remove the chef. When admin clicks on update button it will redirect to update chef template. Where it show all previous details of Chef. Admin will update these previous values and press Update button then all details are validated through jQuery if any error occure then it will show it in alert box and then redirect to same page. If all details are proper then it will show Chef updated successfully. When Admin clicks on delete button it will delete that Chef from database and again redirect to view Chef Page.

### **5.4.5 View order**

All the orders given by the customers will be shown here with orderid, username and totalprice. It also show viewdetail button. When admin clicks on view detail button then it will redirect to view detail Page. In that page description of selected orderid.

### **5.4.6 View Customer**

All the customer details will be shown here within table with username, first name, last name, password, emailed and mobileno.

## 5.5 Client

### **5.5.1 Login and Registration**

Login and Registration facilities are implemented for Customers. We implemented login page for Customers. Login done by username or email id and password. When user clicked on “LOGIN” button after entered username and password at that time in backend side user is validated. If user is validated then in session username, unique orderid will be stored.

In login page there is a link for registration and forgot password.

To register any user, Username, First name, last name, Email ID, Mobile No., and Password required.

### 

### **5.5.2** **Place Order**

User can choose the item and add to the cart.User have to select quantity after adding item to the cart. After confirming the details of the cart user can proceed further for payment. Total price and order id will be displayed after it. Here user can cancel the order if he/she wish. Further user will be redirected to payment gateway for payment process.

### **5.5.3** **History**

Here user can show all the details of item which they have ordered previously. User can also cancel the order from here if it’s not in process or complete.

### **5.5.4 Change Password**

We implemented change password facility. Implementation of change password is done by current password and new password and confirm password after login. When user clicked on change password then first it will compare current password with password which is on database. If both are same then system will check weather new password and confirm password are same or not. If both are same then current password will be replaced by new password.

### **5.5.5** **Forgot Password**

We implemented forget password facility using username and mobile no and email ID as per user details. First system will validate username mobile number and email ID after that system will display the user’s current password.

## 5.6 Chef

### **5.6.1 Manage Order**

Here all the order given by the user will be shown here with its status i.e. pending complete, process. Chef can change this status accordingly. once the order is complete item will remove from the list.

### **5.6.2 History**

Here chef can view the history of the completed order.

## 5.7 Payment

When user clicked on “Buy now” button page is redirected to payment gateway. Before redirecting totalprice, orderid and username is stored in session, so that if transaction will success then system will able to make record in database. Here PAYPAL-API used as gateway.

All the transaction message send into PayPal account. All transaction maintain in back end database.

## 5.8 Exception Handling

When system fetch any details from database then SQLException is used. And to get other exception “Exception” is used.

## 5.9 Logout

We have provide logout facility for all the panel like Admin, Chef and Client. When any user clicks on logout then it will redirect to the respective Login page and delete all the session which are created and active currently.

CHAPTER

# TEST CASE DESIGN

Testing is a process carried out on software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications.

* 1. **Testing Plan:**

The testing technique that is going to be used in the project is Black Box Testing. The expected inputs to the system are applied and only the outputs are checked.

* 1. **Testing Strategy:**

The development process repeats this testing sub process a number of lines for the following phases.

• Unit Testing

• Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification. Testing is carried out in such a hierarchical manner to that each component is correct and the assembly/combination of component is correct. We have performed both Unit Testing and System Testing to detect and fix errors.

* 1. **Testing Methods**

We have performed Black-box testing for the testing purpose. A brief description is given below:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

* 1. **Test Cases:**

**REQ-1.REGISTRATION OF CUSTOMERS**

1. PRE CONDITION : “KANO” USERNAME ALREADY EXISTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO REGISTRATION PAGE | USERNAME: KANO  FIRSTNAME:SAGAR  LAST NAME:PATEL | REGISTRATION  SUCCESULL. | USERNAME IS ALREADY EXIST. | PASS |
| **2** | PROVIDE USER DETAILS | EMAIL ID:SAGARPATEL@GMAIL.COM |  |  |  |
| **3** | CLICK ON REGISTARTION BUTTON | MOBILE NO.: 9876544210  CITY:SURAT  DEPARTMENT:EDUCATIONAL  SALARY:80000  PASSWORD:\*\*\*\*\*\*\* |  |  |  |

1. NO PRE CONDITION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO REGISTRATION PAGE | USERNAME: PINTU  FIRSTNAME:SAGAR  LAST NAME:PATEL | REGISTRATION  SUCCESULL. | REGISTRATION  SUCCESULL. | PASS |
| **2** | PROVIDE USER DETAILS | EMAIL ID:SAGARPATEL@GMAIL.COM |  |  |  |
| **3** | CLICK ON REGISTARTION BUTTON | MOBILE NO.: 9876544210  CITY:SURAT  DEPARTMENT:EDUCATIONAL  SALARY:80000  PASSWORD:\*\*\*\*\*\*\* |  |  |  |

**REQ-2.PAYMENT:**

**1** Invalid payment details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | CHOOSE ORDER | ITEMLIST | PAYMENT  SUCCESSFULLY | PAYMENT PROCESS  ABORTED | PASS |
| **2** | PLACE ORDER | QUANTITY WITH EACH ITEM IN ITEMLIST |  | TRANSACTION INVALID DUE TO IMPROPER CARD DELTAIL. |  |
| **3** | REDIRECT TO PAYMENT | PAYPAL ACCOUNT LOGIN CREDETIALS |  |  |  |
| **4** | PROVIDE  PAYMENT  INFORMATION | CARD TYPE  CARD HOLDER NAME  CARD NUMBER  PASSWORD  CVV |  |  |  |
| **5** | CLICK PAYMENT BUTTON |  |  |  |  |

**REQ-3.VIEW ITEM/CHEF/ORDER/CUSTOMER/ORDERDETAILS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | SELECT BUTTON AS PER VIEW |  | VIEW OF SELECTED FIELD | VIEW OF SELECTED FIELD | PASS |

**REQ-4.HISTORY NOTE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | VIEW  HISTORY |  | DISPLAY  HISTORY | DISPLAY  HISTORY | PASS |

**REQ-5.LOGIN**

1 INVALID USERNAME OR PASSWORD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO LOGIN PAGE | USERNAME:  KANO  PASSWORD:  \*\*\*\* | LOGIN SUCCESSFUL. | INVALID UERNAME OR PASSWORD | PASS |
| **2** | PROVIDE LOGIN DETAILS |  |  |  |  |
| **3** | CLICK ON LOGIN BUTTON |  |  |  |  |

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO LOGIN PAGE | USERNAME:  KANO  PASSWORD:  \*\*\*\*\*\* | LOGIN SUCCESSFUL. | LOGIN SUCCESSFUL | PASS |
| **2** | PROVIDE LOGIN DETAILS |  |  |  |  |
| **3** | CLICK ON LOGIN BUTTON |  |  |  |  |

**REQ-6. CHOOSE ORDER**

1 PRE CONDITION: CUSTOMER LOGIN SUCCESFUL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO CHOOSE ORDER PAGE | ITEMS NOT ADDED | ORDER SUCCESSFUL. | NO ITEM SELECTED | PASS |
| **2** | PROVIDE ITEM LIST  CATEGORY NOT SELECTED |  |  | NO QUANTITY SELECTED. | PASS |
| **3** | CLICK ON CART BUTTON |  |  |  |  |

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO SPENDMONEY PAGE | ITEMS ADDED | ORDER SUCCESSFUL. | ORDER  SUCCESSUL | PASS |
| **2** | PROVIDE ITEM LIST  CATEGORY NOT SELECTED |  |  |  |  |
| **3** | CLICK ON CART BUTTON |  |  |  |  |

**REQ-7.CHANGE PASSWORD:**

1 PRE CONDITION: LOGIN SUCCESSFUL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO CHANGE PASSWORD PAGE | CURRENT PASSWORD:  \*\*\*\*\*\*  NEW PASSWORD:  \*\*\*\*\*\*\*\* | CHANGE PASSWORD SUCCESSFUL | INVALID CURRENT PASSWORD | PASS |
| **2** | PROVIDE NEW DETAILS | CONFIRM PASSWORD:  \*\*\*\*\*\*\*\* |  | NOT MATCH CONFIRM PASSWORD | PASS |
| **3** | CLICK ON CHANGE PASSWORD BUTTON |  |  |  |  |

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO CHANGE PASSWORD PAGE | CURRENT PASSWORD:  \*\*\*\*\*\*  NEW PASSWORD:  \*\*\*\*\*\*\*\* | CHANGE PASSWORD SUCCESSFUL | CHANGE PASSWORD SUCCESSFUL | PASS |
| **2** | PROVIDE NEW DETAILS | CONFIRM PASSWORD:  \*\*\*\*\*\*\*\* |  |  |  |
| **3** | CLICK ON CHANGE PASSWORD BUTTON |  |  |  |  |

**REQ-8.FORGET PASSWORD:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO FORGET PASSWORD PAGE | USERNAME:  KANO  EMAILID:vivekbhingaradiya258@gmail.com  MOBILE NO:9876544210 | PASSWORD SEND SUCCESSFUL | INVALID ENTERED  DATA | PASS |
| **2** | PROVIDE USERNAME EMAILID AND MOBILE NO |  |  |  |  |
| **3** | CLICK APPLY BUTTON |  |  |  |  |

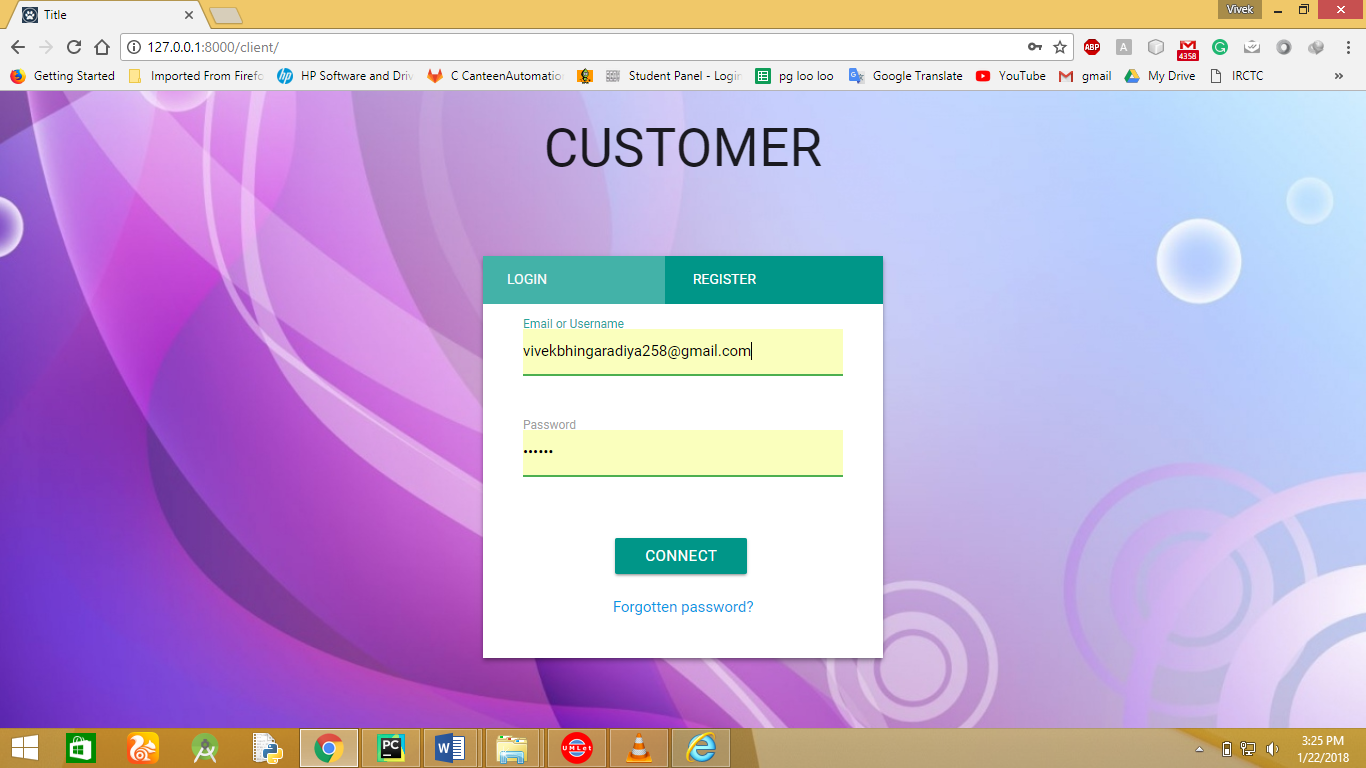
2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STEP** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** | **PASS/FAIL** |
| **1** | REDIRECT TO FORGET PASSWORD PAGE | USERNAME:KANO  MOBILE NO:9876543210 | PASSWORD SEND SUCCESSFUL | PASSWORD SEND SUCCESSFUL AND LOGIN WITH NEW PASSWORD | PASS |
| **2** | PROVIDE USERNAME AND MOBILE NO |  |  |  |  |
| **3** | CLICK APPLY BUTTON |  |  |  |  |

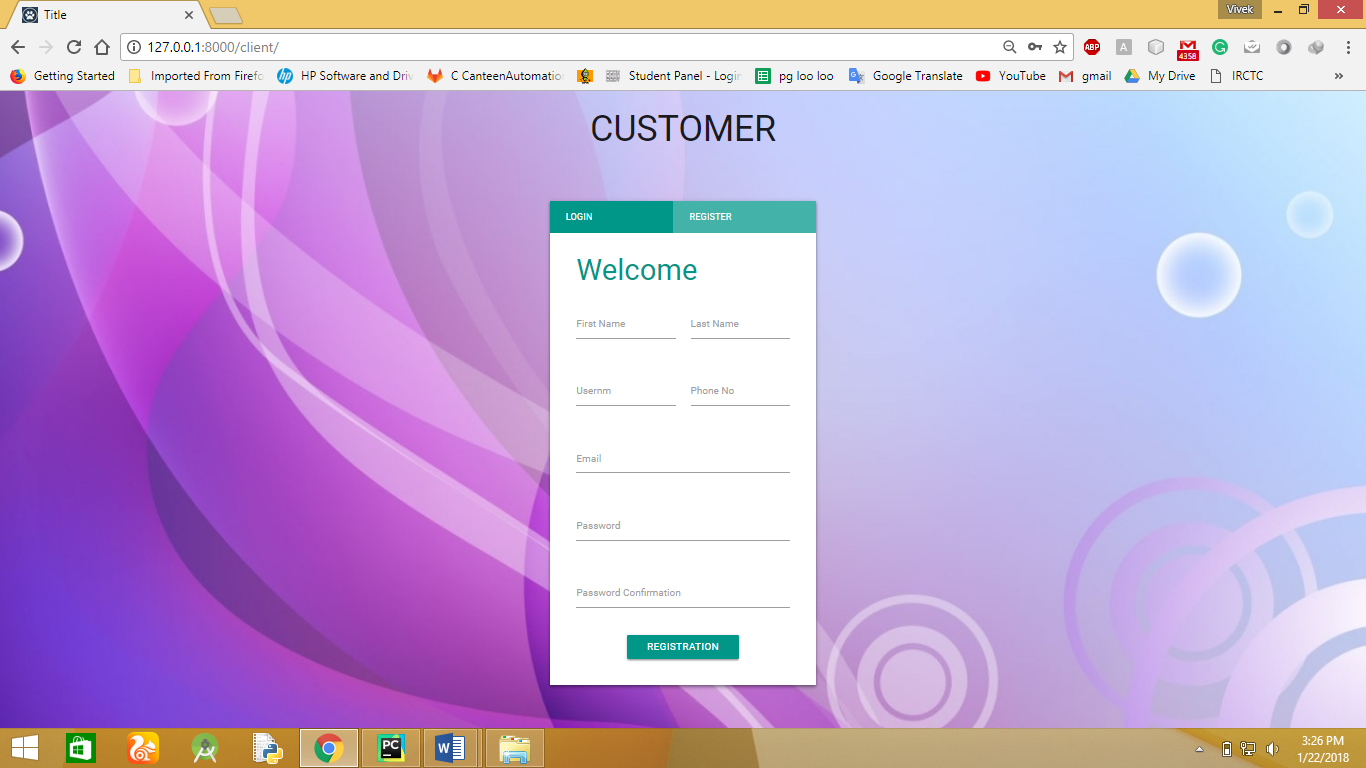
CHAPTER

# SCREENSHOTS

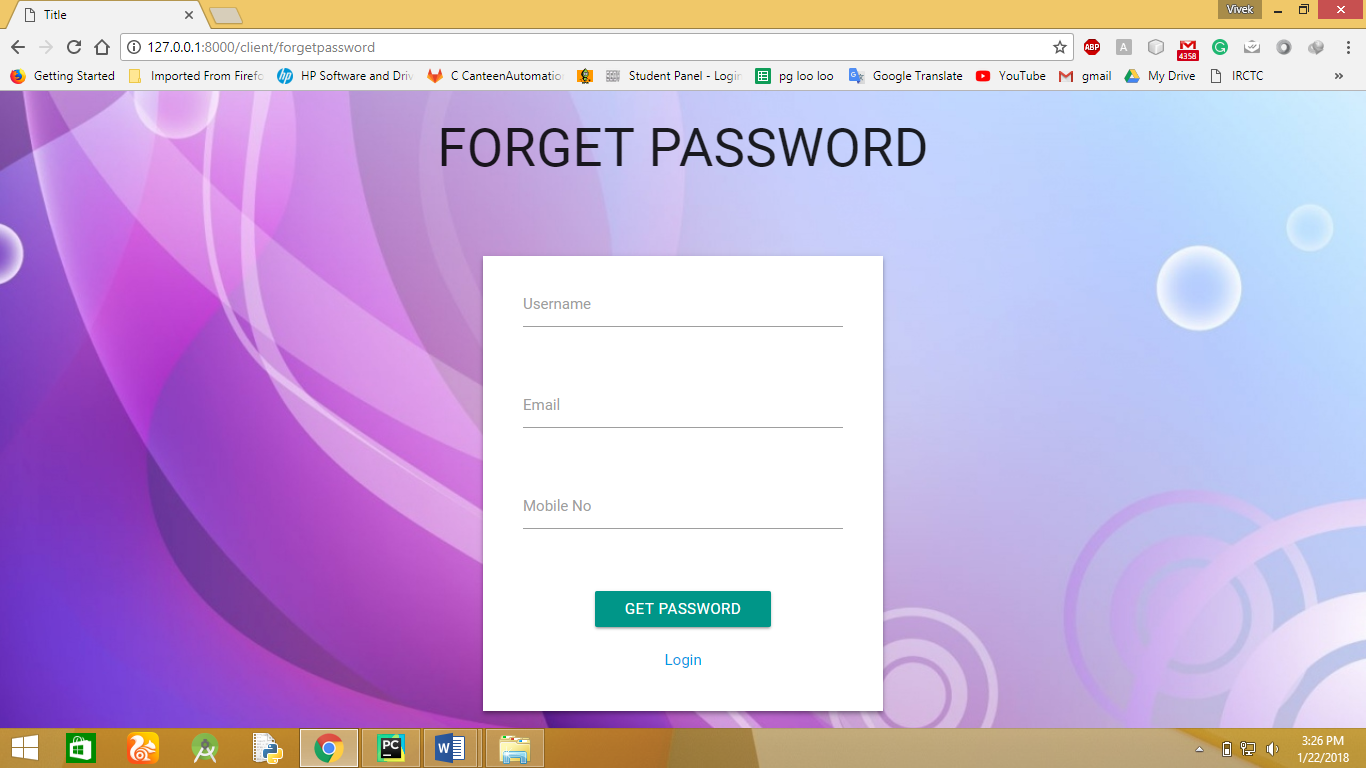
**CUSTOMER LOGIN**

****

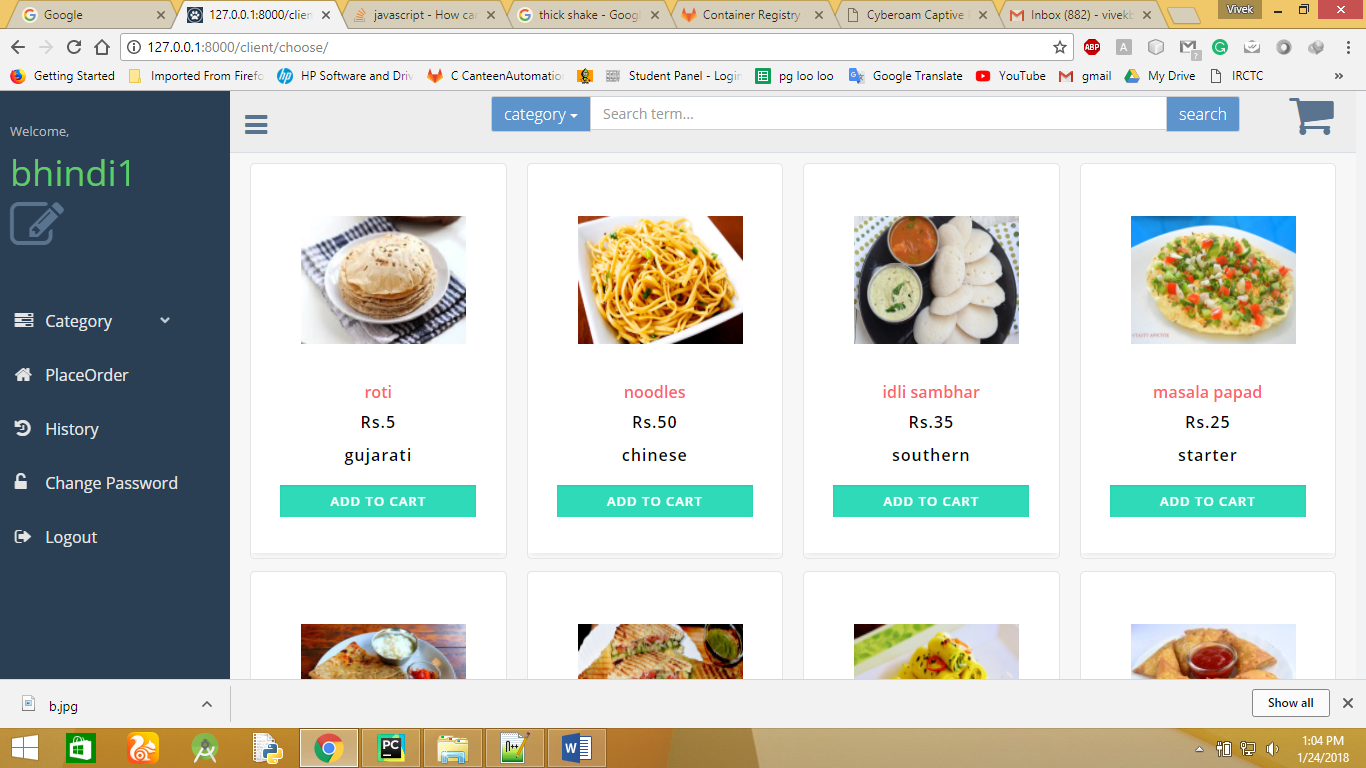
**CUSTOMER REGISTRATION**

****

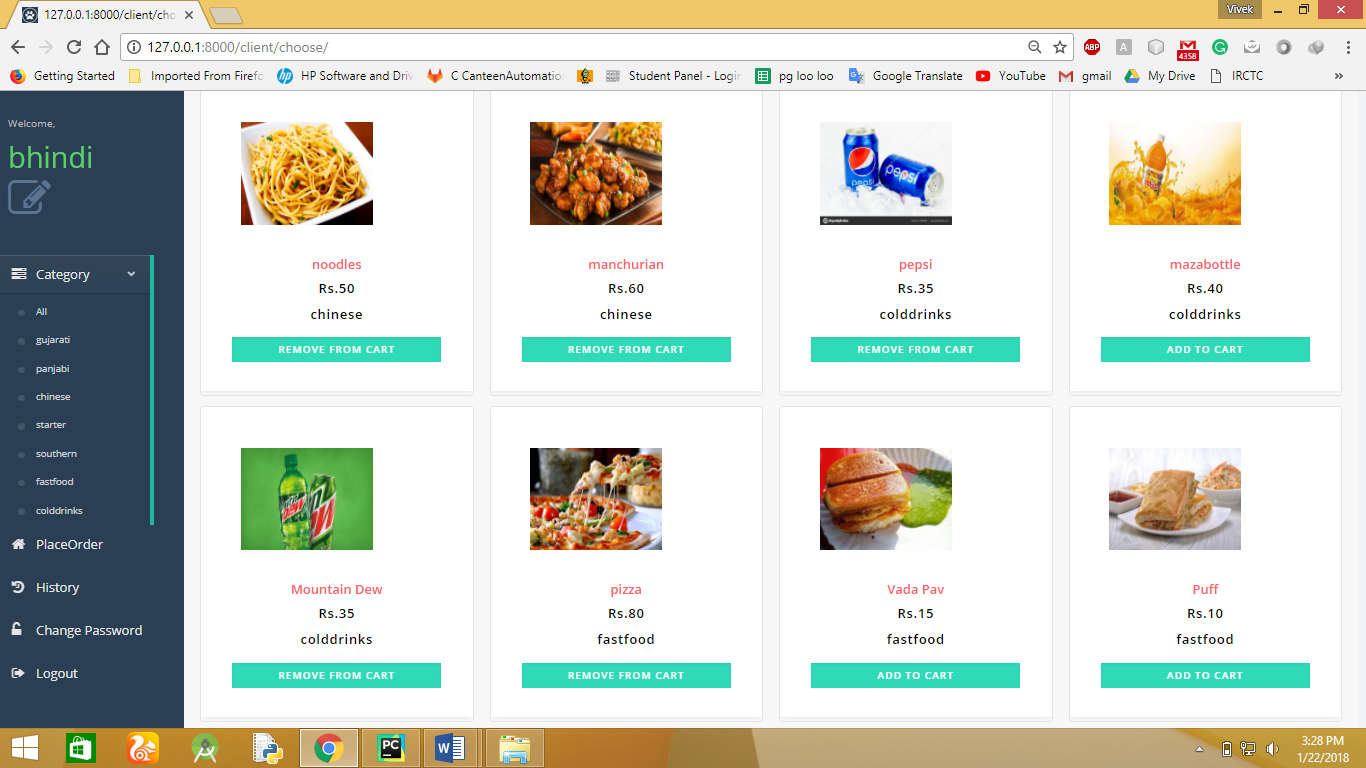
**FORGET PASSWORD**

****

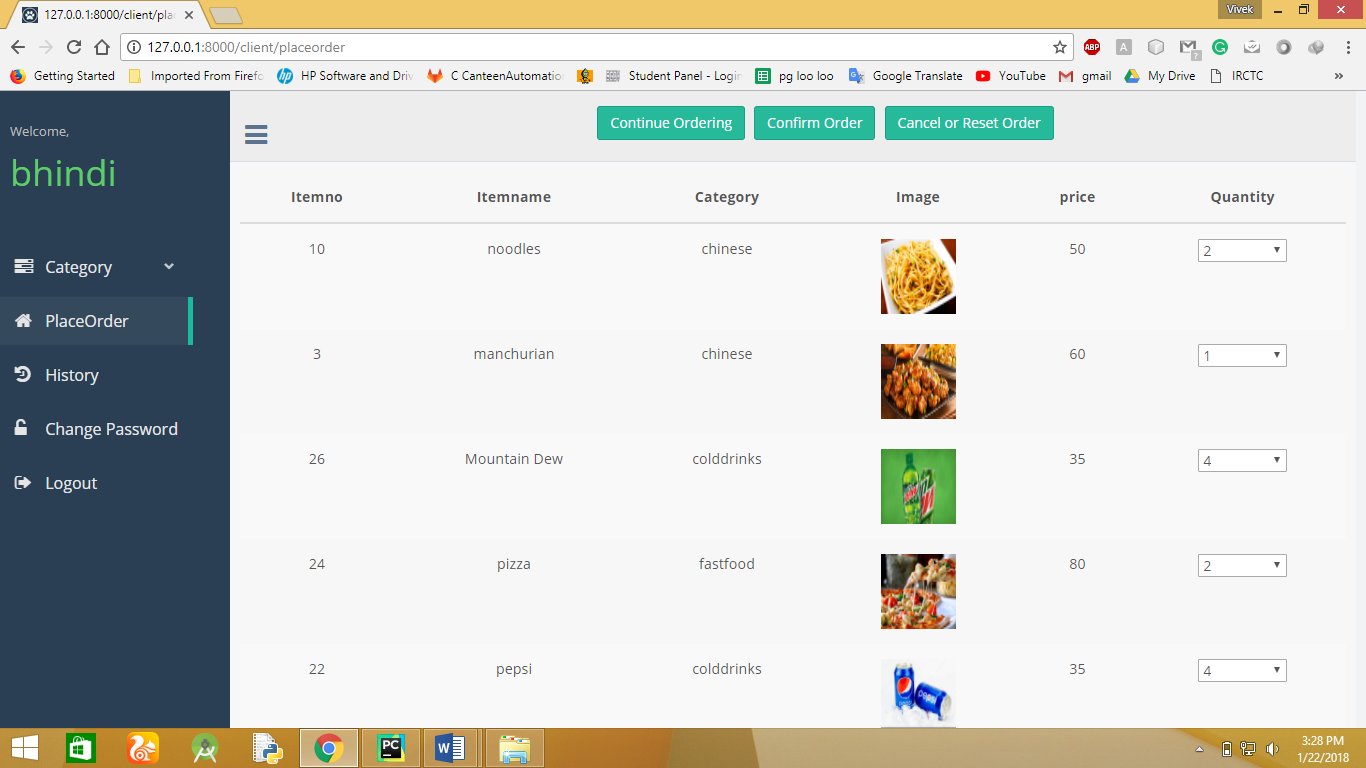
**CHOOSE ORDER**

****

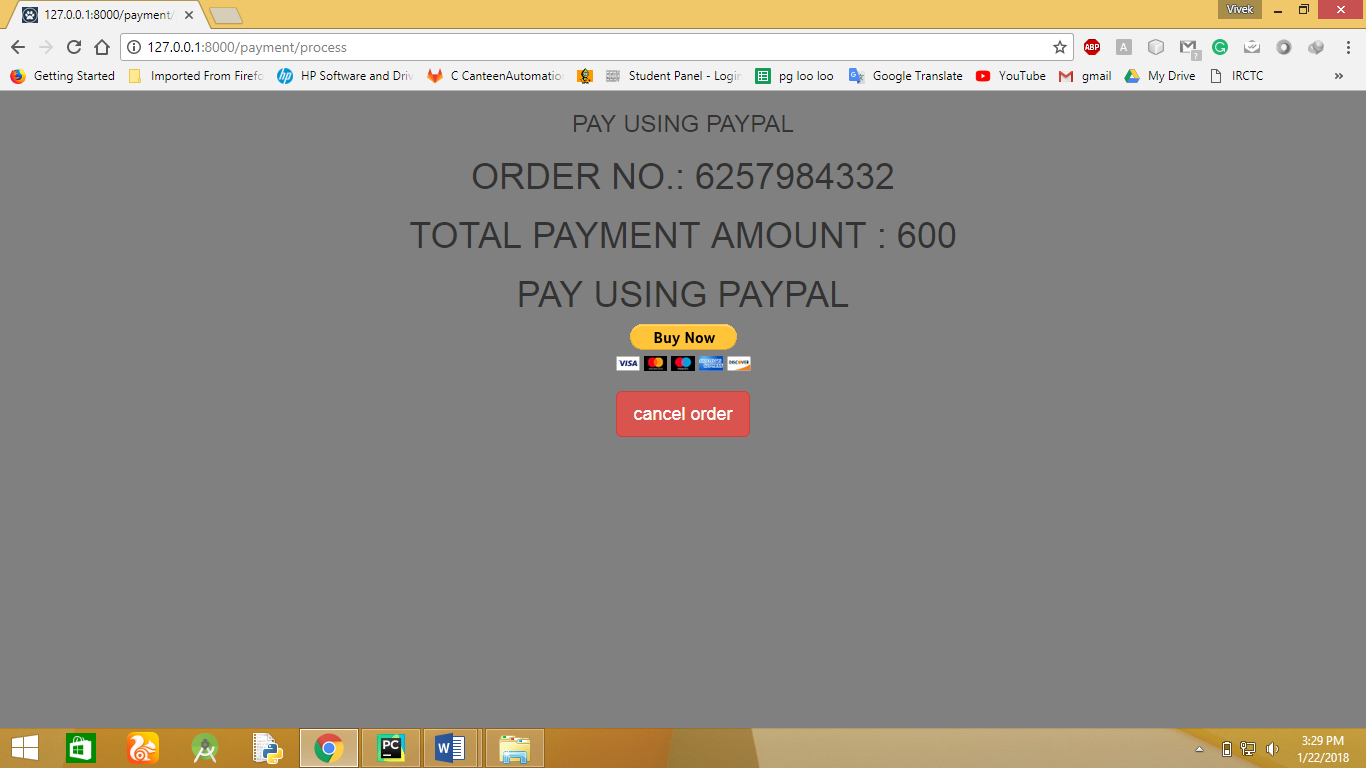
## ADD TO CART

****

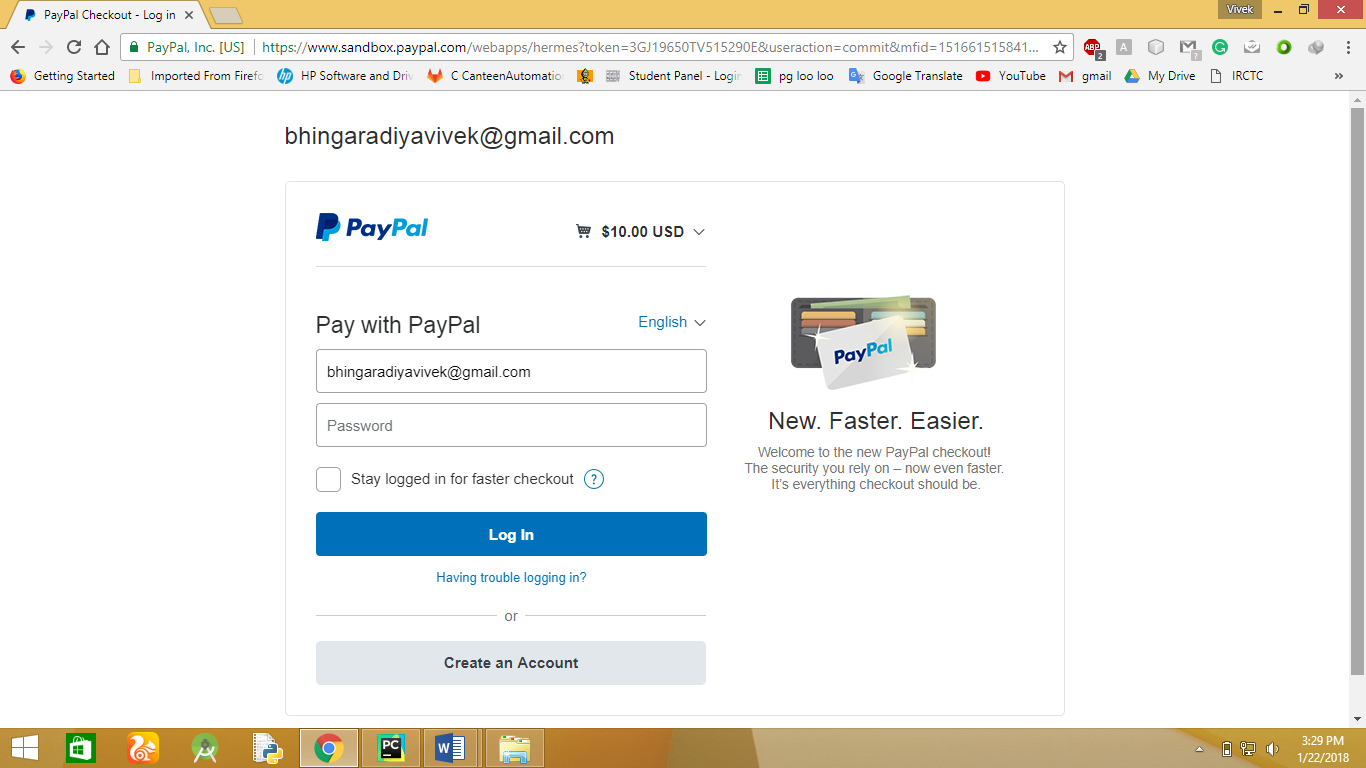
**PLACE ORDER**

****

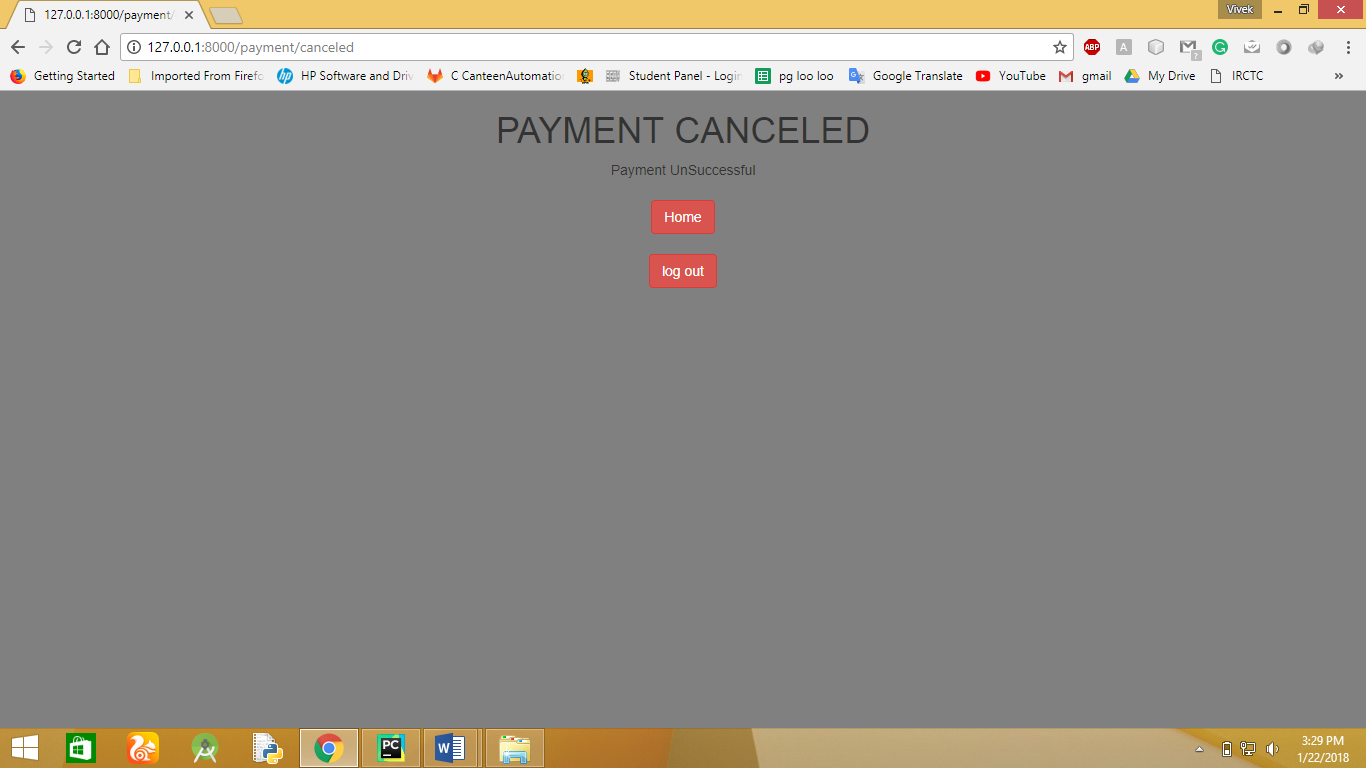
**CONFIRM ORDER**

****

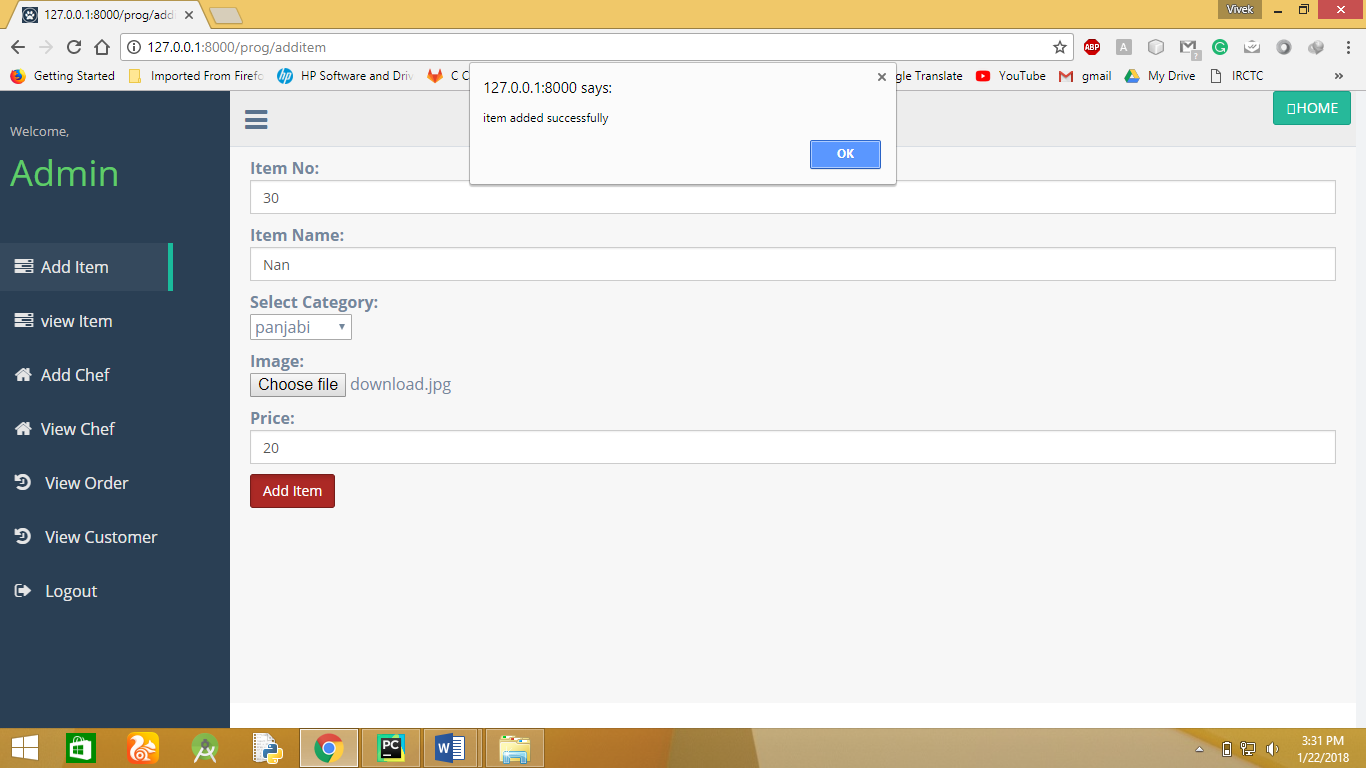
**PAYPAL PAYMENT**

****

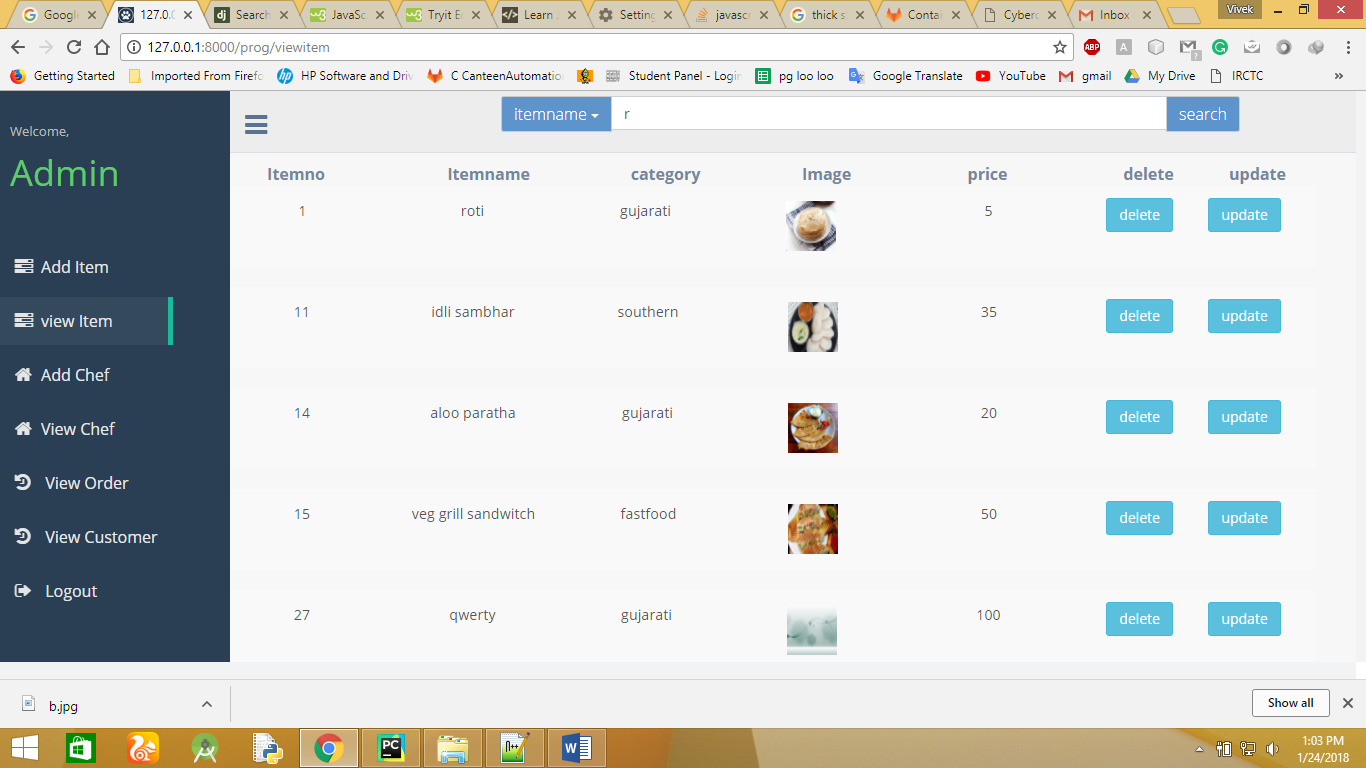
**CANCEL PAYMENT**

****

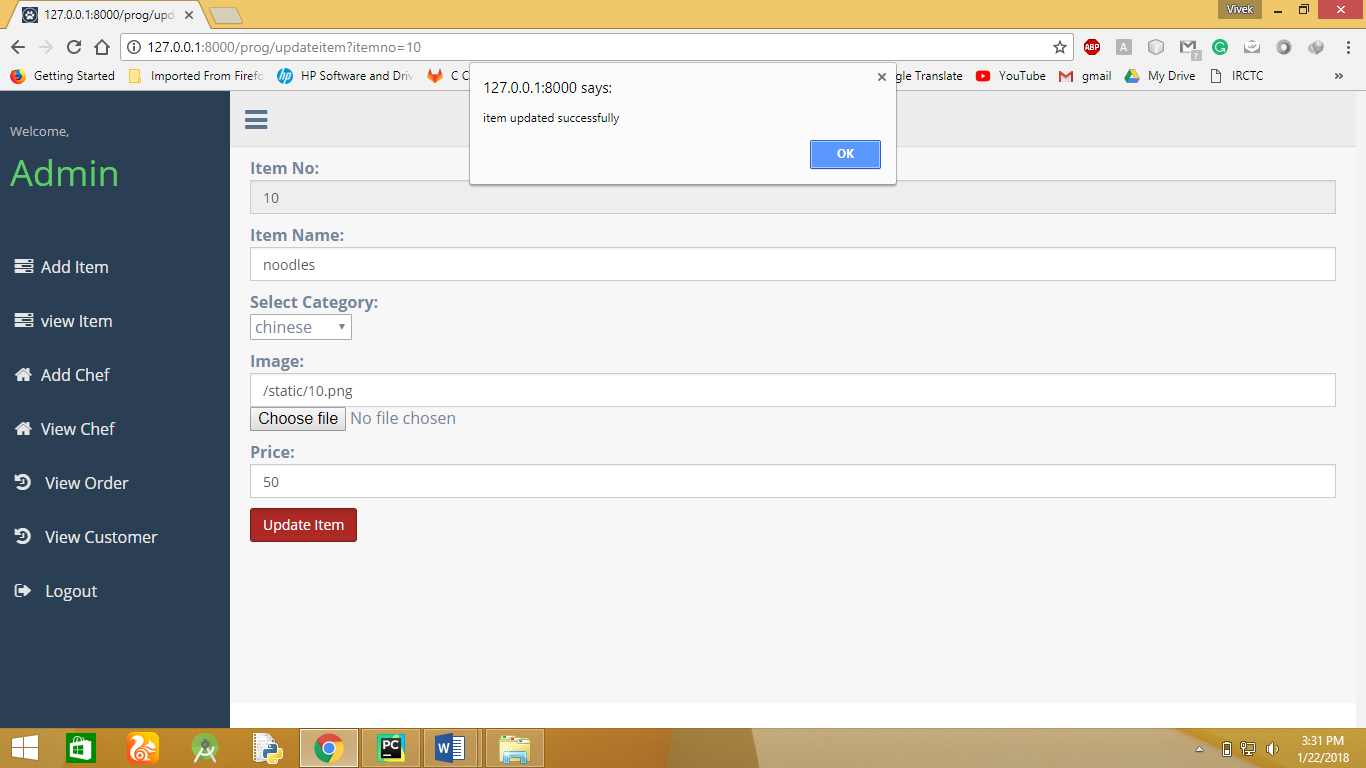
**ADD ITEM**

****

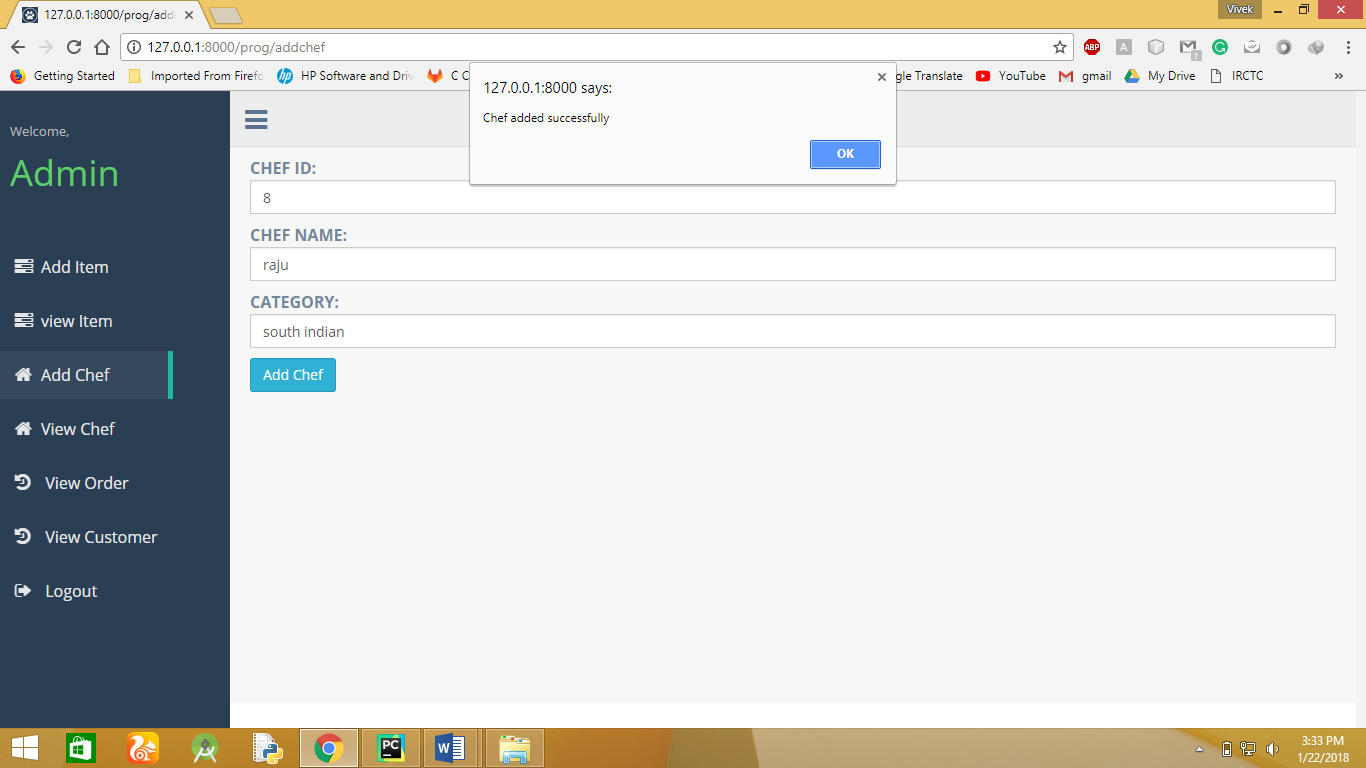
**VIEW ITEM**

****

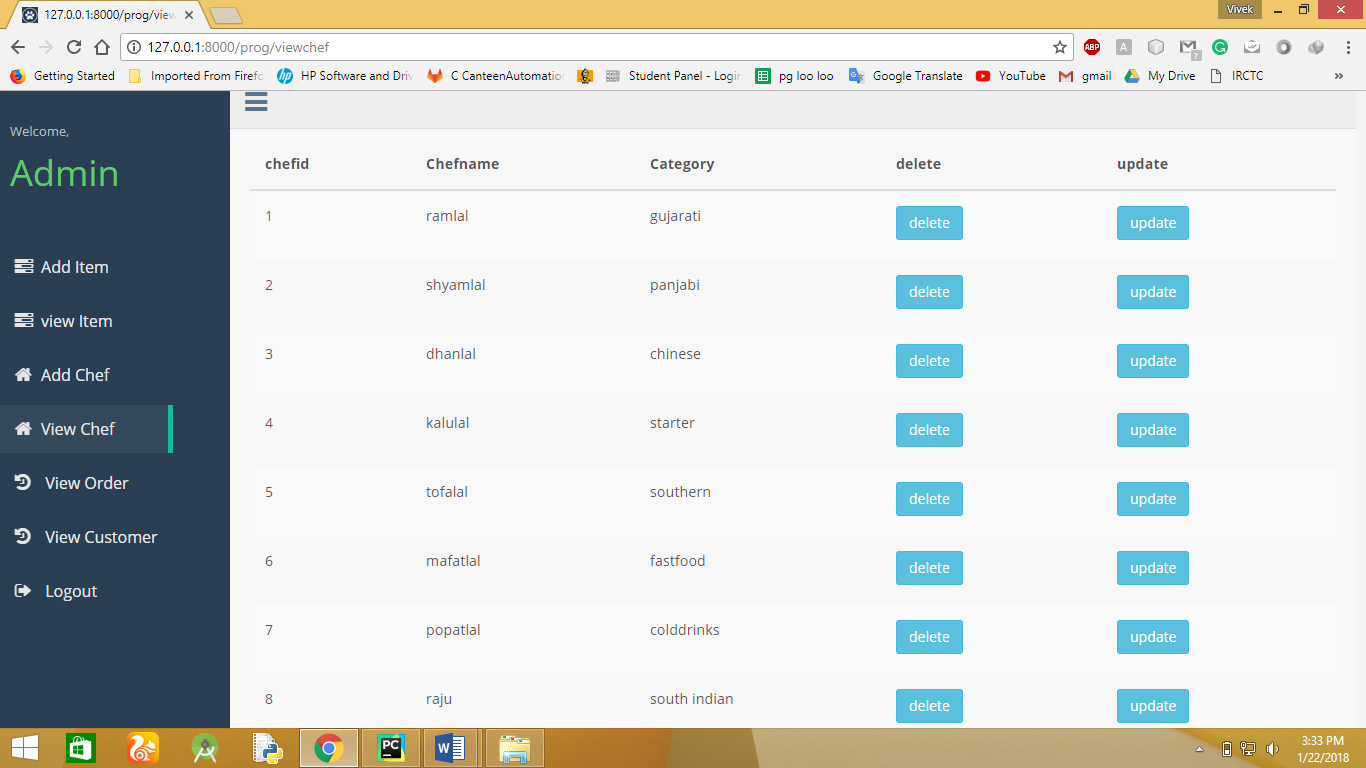
**UPDATE ITEM**

****

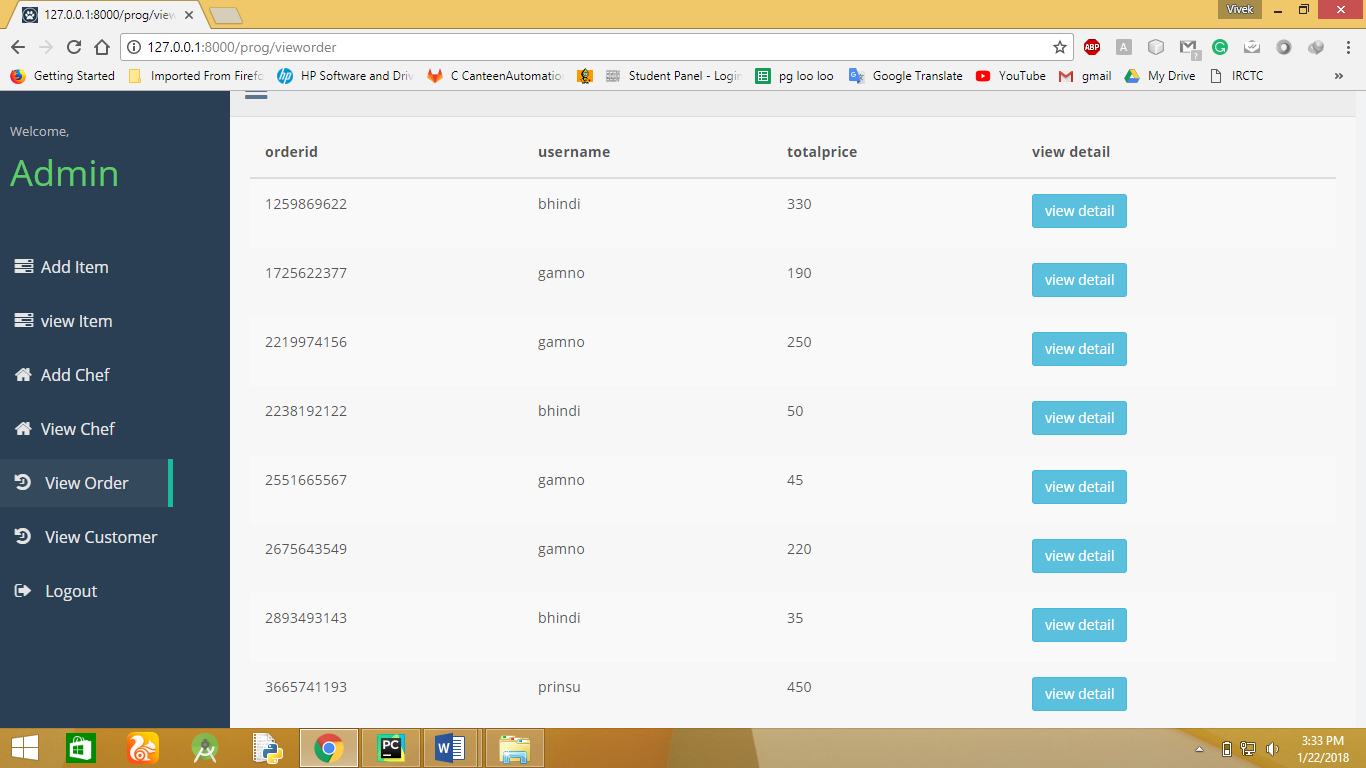
**ADD CHEF**

****

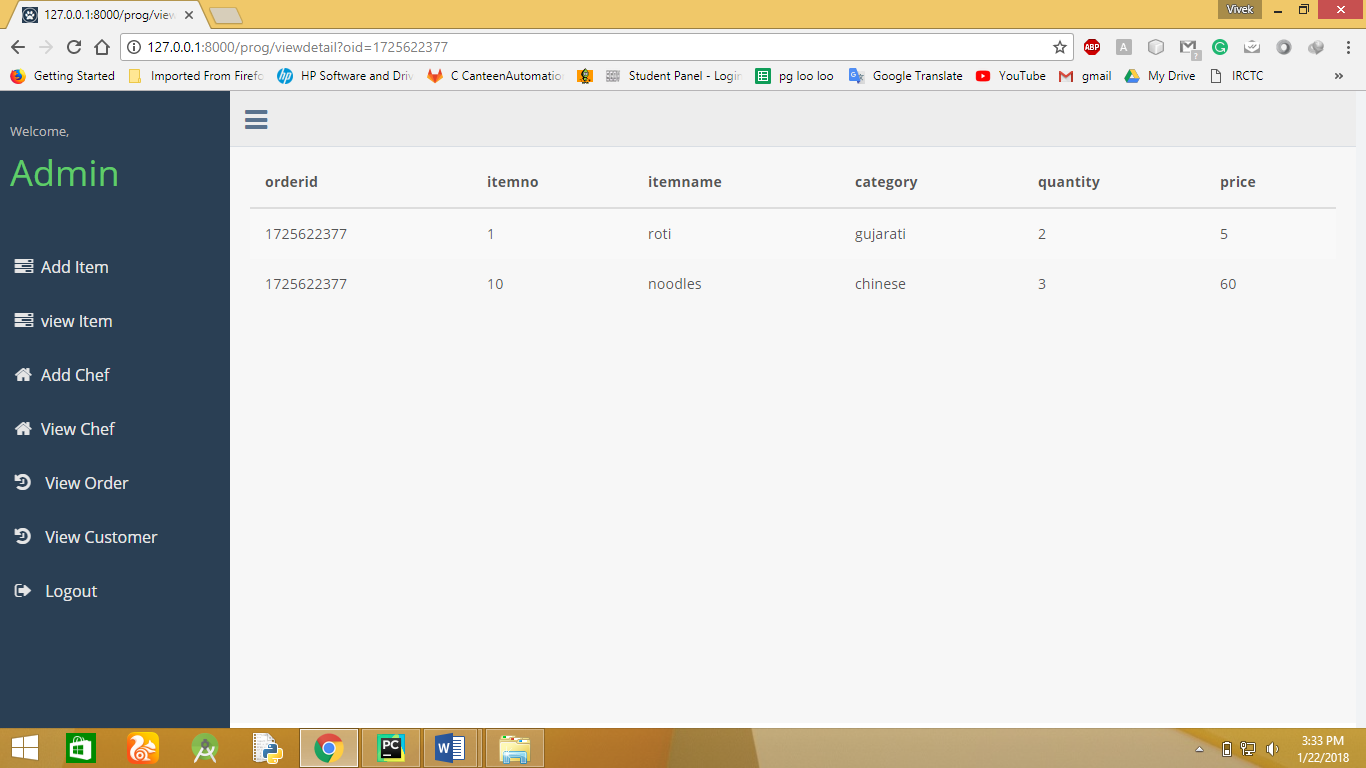
**VIEW CHEF**

****

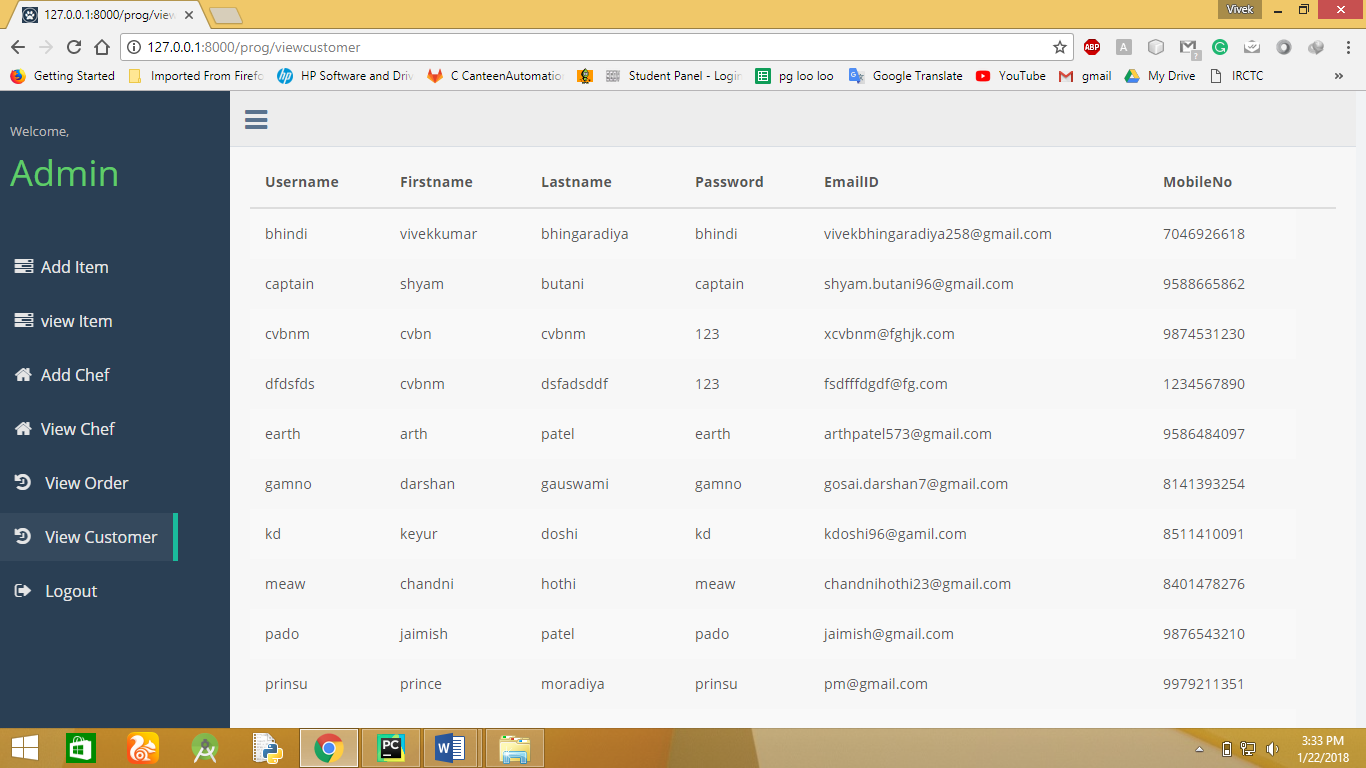
**VIEW ORDER**

****

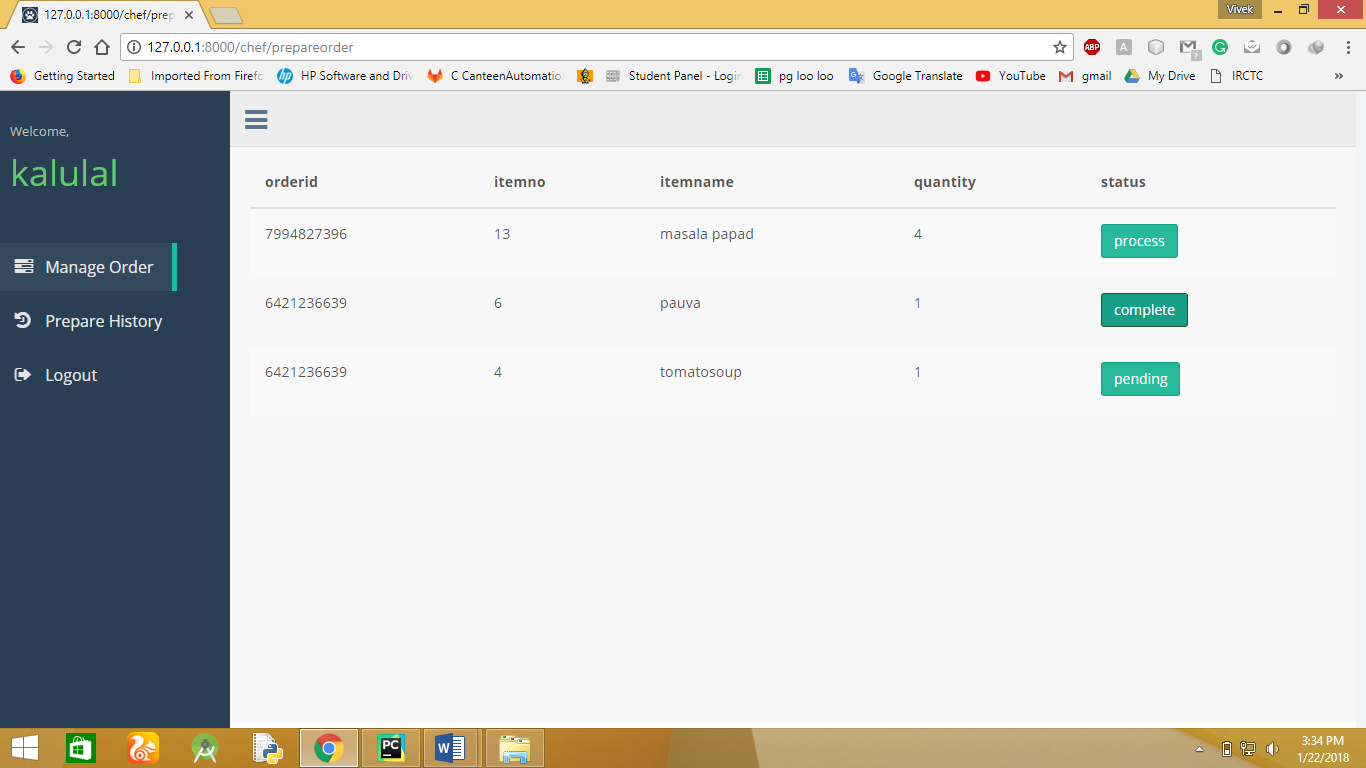
**VIEW ORDERDETAILS**

****

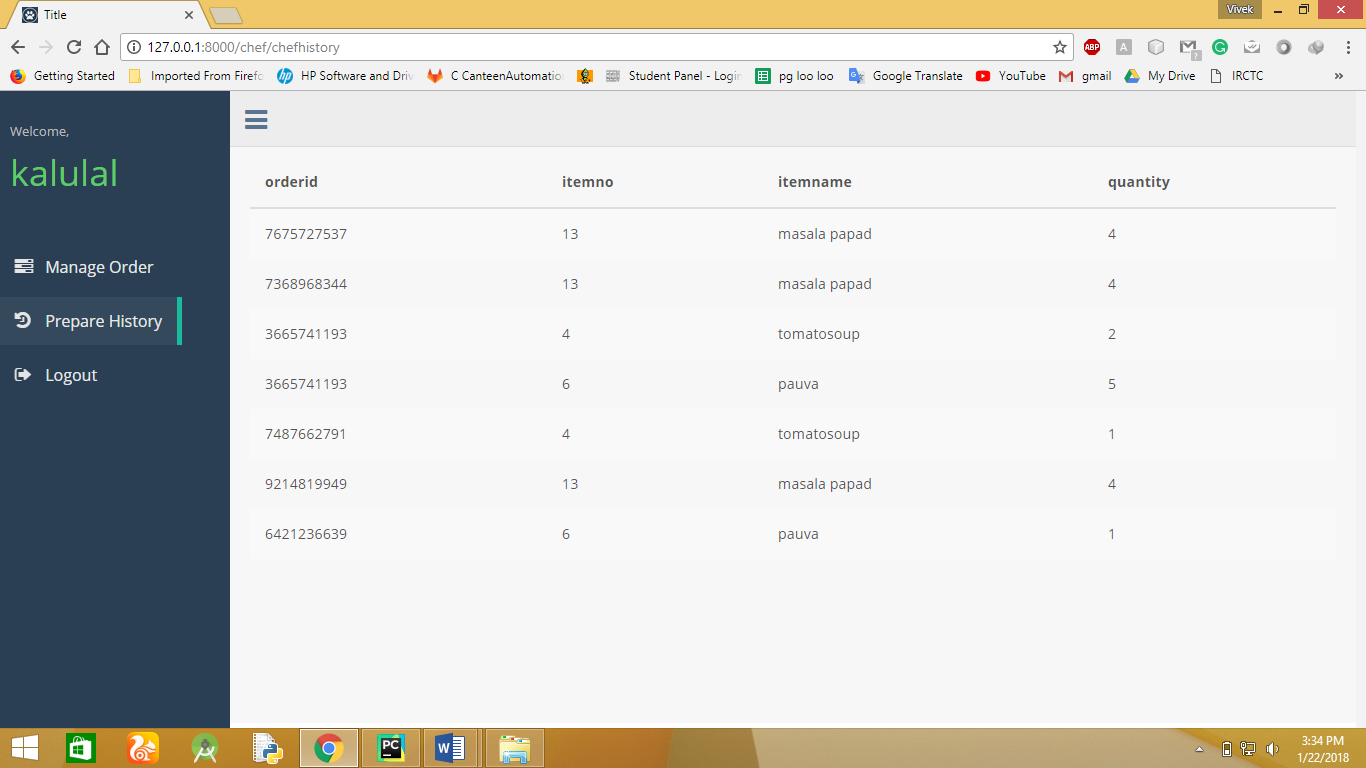
**VIEW CUSTOMER**

****

**MANAGE ORDER**

****

**PREPARE HISTORY**

****

**CHAPTER**

1. **CONCLUTION AND FUTURE EXTENTION**

**CONCLUSION:**

Hereby we declare that, we have performed a project by understanding all the modules of this project. We have checked the feasibility and the requirements of the system. Then we designed the GUI and the flow control roughly on the paper. After this the actual implementation of our modules was started. Initially all the modules were developed separately and lately all were integrated with each other.

Next phase we came across was testing. We tested each module individually and later the whole system. And thus through this system, an organization can easily maintain records of all the users efficiently and can perform various operations easily.

We experienced an enhancement in our Designing and technical skills of programming through the development of this Software.

**LIMITATION:**

* The software doesn’t point at the errors when the user types the wrong text.
* When there is a problem with internet speed, it might create problem for forget password and payment.
* Here we have used Paypal API, so that if its server down then it will affect the performance of this project.

**FUTURE EXTENSION**:

* Add more functionality into this system.
* Also Original Payment Integration.
* Use of Apache Solr for searching functionality, because it is much faster.

**CHAPTER**

1. **BIBLOGRAPHY**

**Books:**

* Fundamental of Software engineering - 2nd Edition by

Rajib Mall, published by – McGrow Hill

* [www.StakeOverflow.com](http://www.StakeOverflow.com)
* [www.oracle.com/database](http://www.oracle.com/database)
* <https://www.tutorialspoint.com/python/>
* <https://tutorial.djangogirls.org/en/>
* <https://docs.djangoproject.com/en/>
* <https://django-paypal.readthedocs.io/en/stable/>