**Logo

Description automatically generated**

**UNIVERSITY OF RAJSHAHI**

**DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING**

**Online FOOD-Order management system for ICEFoods**

*Submitted to*

**Dr. Dipankar Das**

**Professor, Dept. of Information and Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh.**

**Dr. Riaz Uddin Mondal**

**Associate Professor, Dept. of Information and Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh.**

**Dr. Mousumi Haque**

**Associate Professor, Dept. of Information and Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh.**

BY

**MD. AHASAN HABIB**

**ID: 1710677142**

**A Group Project Submitted in Partial Fulfillment of Lab assessment**

**March, 2023**

**1.1 Introduction**

This manuscript is proposing about all the features and dealings to develop the system. Especially it is containing details about objectives, possibility, plan replica, primary and functional requirements, database model and as final point coverage and analyzing the mechanisms. Once analyzing the mechanisms of the task that would be performed, the following point is to consider the problem and understand is framework. Online Restaurant Management System is a project which is referred to as a set of detail methods that is being used in handling the ordering process. Food ordering can be computerized or done manually. Those helps the customer to order their food themselves which is known as the customer self ordering system. The customer self-ordering system can be defined as a computerized system that is being used by customers to place their own orders in the restaurant and allow the orders to be tracked, in order to prepare and deliver the food to the computers. Admin is the most powerful user of the system

**1.2 What is Online Restaurant Management System?**

Online restaurant is a process of ordering food from a local restaurant or food cooperative through a web page. Much like ordering consumer goods online, many of these allow customers to keep accounts with them in order to make frequent ordering convenient. A customer will search for a favorite restaurant, usually filtered via type of cuisine and choose from available items, and choose delivery or pick-up. Payment can be amongst others either by credit card or cash, with the restaurant returning a percentage to the online food company.

**1.3 Background of the project**

The online restaurant management 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 cheque. 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 were placed over the phone, but there are many disadvantages to this system. It is possible for anybody to order any goods via the internet and have the goods delivered at his/her doorsteps. But while trying to discuss the transfer method of the goods and services, attention is focused on the payment mode. In other words, how possible is it to pay for goods and services via the 3 internet? This then leads to the discussion of the economic consequences of digital cash. What I propose is an online ordering system originally designed for use in college cafeterias, but just as applicable in any food delivery industry. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and the restaurant. The system also greatly lightens the load on the restaurants end, as the entire process of taking orders is automated. Once an order is placed on the webpage that will be designed, it is placed into the database and then retrieved, in pretty much real-time, by a desktop application on the restaurants end. The greatest advantage of this system is its FLEXIBILITY .

**1.4 Objective of the project**

This project Online restaurant aim 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.

**2.1 Requirement Engineering**

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. Requirements analysis is critical to the success or failure of a systems or software project.

The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

**2.2 System Requirement**

Our system can be used in windows 7, 8, 10 and 11 with 32 bit, and 64 bit operating system and also supported for other platform such as Linux OS X.

**2.3 Software and Hardware Requirements**

**2.3.1 Software Requirements:**

* Web server software: Apache
* Server side scripting tool: PHP
* Database tools: MYSQL DBMS
* Compatible operating system: Windows, Mac
* Front End design tools: html5, CSS, JavaScript, J Query

**2.3.2 Hardware Requirements:**

* Hardware recommend by all the software needed.
* RAM : 256MB or more
* Hard Drive: 10GB or more

**2.4 User Requirements**

To deliver the best service to the users we tried to find out the user necessities which are below:

**Administrator Aspect:**

* Monitoring the whole system from admin panel.
* Taking back up of the database.
* Creating, deleting and modifying the records.
* Add users for the admin panel.
* Add customers and other staff.
* Keeping the customer’s record like their details.
* Organizing their member registration system.
* Approve the notice to post.
* Monitoring the transaction system.

**Customer Aspects:**

* Signing in and signing up to the system.
* Changing their password.
* Resetting forgot password

**2.5 Functional Requirements**

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do ", while non-functional requirements are "system shall be ". The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system

**2.6 Non-Functional Requirements**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

**2.7 Business Requirements**

Business requirements in the context of software engineering or the software development life cycle, is about eliciting and documenting business requirements of business users such as customers, employees, and vendors early in the development cycle of a system to guide the design of the future system. Business requirements are often captured by business analysts, who analyze business activities and processes, and often study As-is process to define a target To-be process.

Business requirements often include

* Business context, scope, and background, including reasons for change
* Key business stakeholders that have requirements
* Success factors for a future/target state
* Constraints imposed by the business or other systems
* Business process models and analysis, often using flowchart notations to depict either 'asis' and 'to-be' business processes
* Logical data model and data dictionary references
* Glossaries of business terms and local jargon
* Data flow diagrams to illustrate how data flows through the information systems (different from flowcharts depicting algorithmic flow of business activities)

**2.8 Data and Category Requirements**

There are dissimilar classes of users namely admin, customer and other staff. Depending upon the category of users the access rights are obvious. It means if the user is an administrator then he/she can be able to adjust the data delete, add etc. All other users expect the restaurant only have the rights to save the info about database. The database stores the detail of customer’s proper time. Admin should be able to update restaurant records.

**2.9 Constraints**

* Deliverable: March 06, 2023.
* Available Budget: N/A
* Delivering for a traditional restaurant with simplest computer infrastructure with two desktops, one laptop, one tablet, and one smartphone
* The system may not be able to let restaurant staff to communicate with delivery personnel nor customers in real-time.

**3.1 Use Case**

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal

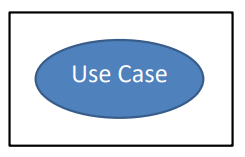
****

Figure 3.1: Use-Case

**3.1.1 Elements of Use Case Diagram**

**Actor:** An actor in the Unified Modeling Language (UML) specifies a role played by a user or any other system that interacts with the subject.

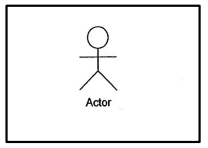
****

Figure 3.2: Actor

**Association:**

An association between an actor and a use case indicates that the actor and the use case somehow interact or communicate with each other.

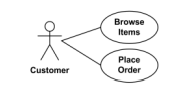


Figure 3.3 : Associations in Use-Case

**Includes:**

Include is used to extract use case fragments that are duplicated in multiple use cases. The included use case cannot stand alone and the original use case is not complete without the included one. This should be used sparingly an only in cases where the duplication is significant and exists by design (rather than by coincidence)

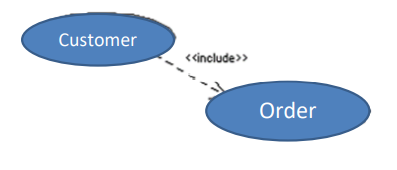


Figure 3.4: Include Relation in Use-Case

**3.1.2 Actor Depiction**

Actors are exterior entities that cooperate with the structure. Actor pledges system activities for the determination of finishing some task. Actors in this project are as follows:

Admin: Monitor the system, add product, add customer, add others staff etc.

Customer: Signing in and signing up to the system, changing their password and resetting forgot password.

**3.1.3 Use-Case Diagram for Online FOOD-Order management system for ICEFoods**

Online food ordering has become an increasingly popular trend, as customers look for convenience, speed, and ease of use when it comes to ordering food. To meet these demands, online food ordering systems have been developed to make the process of ordering food online as easy and convenient as possible. One key tool for developing such systems is the use case diagram, which helps to identify the actors involved in the system and the specific use cases that they require.

In this portion, we will examine the use case diagram for an online food restaurant system that includes customers as the primary actors, with additional secondary actors such as the admin, employee, and bank. We will also look at the use cases for each actor, including customers, admin, employee, and bank.

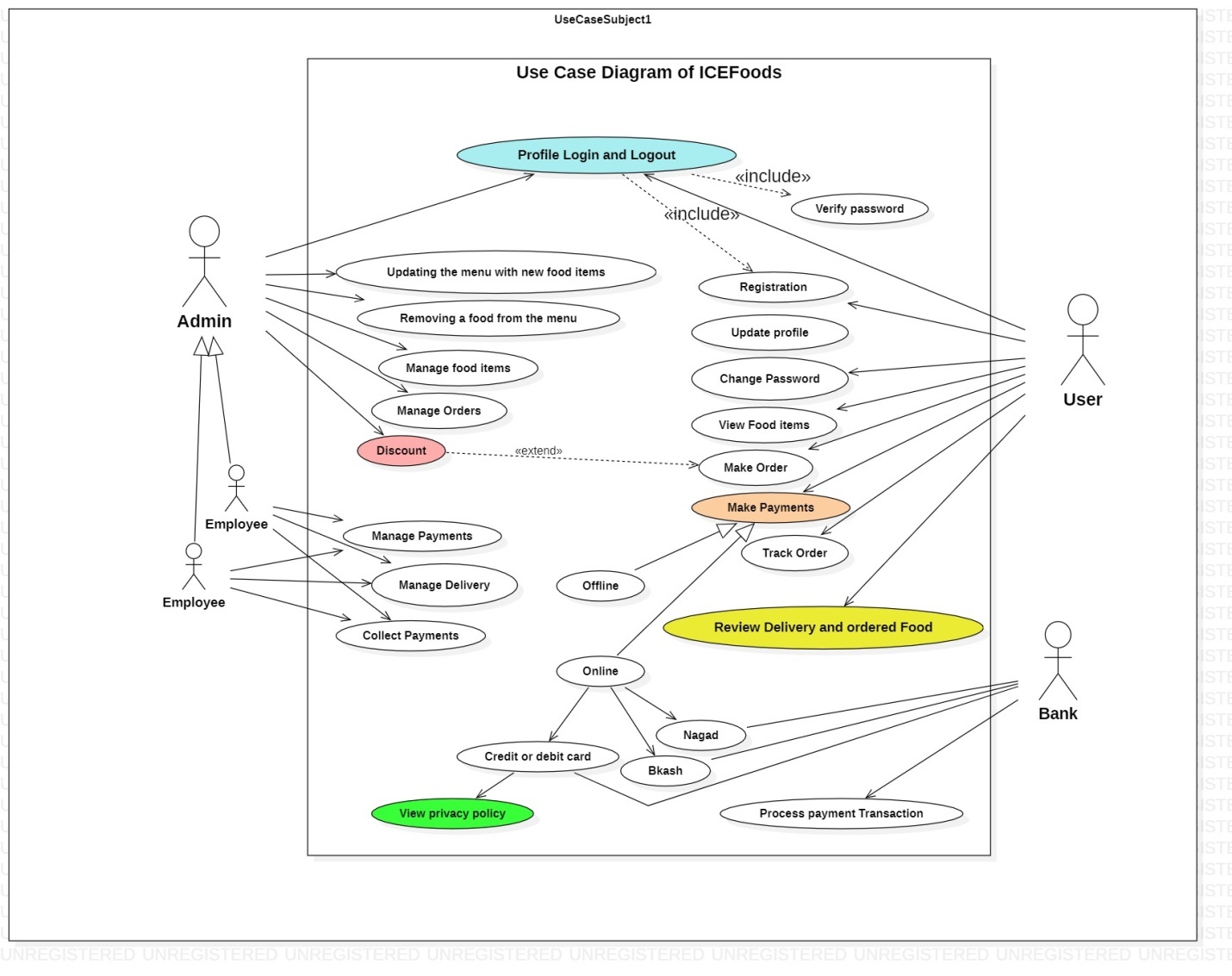


Figure 3.5: Use Case Diagram.

The use case diagram for the online food restaurant system includes several primary actors, including customers, and several secondary actors, including the admin, employee, and bank. The use cases for each actor are as follows:

**Customers**

* Registration and update profile: This use case allows customers to register with name, email, password and contact view and edit their profile information, including name, address, and contact details.
* Login and Logout: This use case allows customers to log in to the system to access their account and log out when they are done.
* Change Password: This use case allows customers to change their password if they forget it or wish to update it for security reasons.
* View Food Items: This use case allows customers to browse the restaurant's menu and view the food items available for order.
* Make Order: This use case allows customers to place an order for food items from the restaurant's menu.
* Make Payment: This use case allows customers to pay for their order using their preferred payment method, such as a credit card or online payment system.
* Track Order: This use case allows customers to track the progress of their order, including its status and estimated delivery time.

Review Delivery and Ordered Foods: This use case allows customers to review the delivery of their order and the quality of the food they ordered.

**Admin**

* Login and logout: This use case allows the admin to log in to the system to access their account and manage the system.
* Update Menu: This use case allows the admin to update the restaurant's menu, adding new food items or removing existing ones.
* Manage Food: This use case allows the admin to manage the food items available for order, including their prices, descriptions, and images.
* Manage Orders and Discounts: This use case allows the admin to manage the orders received through the system, including their status and any discounts applied to them.

**Employee**

* Manage Payment: This use case allows the employee to manage the payment process, including confirming payment and handling any issues that may arise.
* Manage Delivery: This use case allows the employee to manage the delivery process, including coordinating with the delivery driver and ensuring that the order is delivered on time.
* Collect Payment: This use case allows the employee to collect payment from customers who have chosen to pay in cash.

**Bank**

* Process Payment: This use case allows the bank to process payments made by customers through the online food restaurant system.

The use case diagram is an essential tool for developing an online food restaurant system that meets the needs of its users. By identifying the primary and secondary actors involved and the specific use cases they require, the system can be developed to meet these needs effectively. The use cases identified for customers, admin, employee, and bank in this report demonstrate the various functions required in an online food restaurant system, and how each actor plays a crucial role in the overall system.

4 SIGNIFICANT ANALYSIS

4.1 User Interface

ICEFoodsshould have separate interfaces for different types of users. The admin interface should have various functionalities, which will be organized and accessible through a navigation bar. The admin user can choose to expand or shrink this navigation bar as per their convenience. The functionalities for Customers and Users will be almost the same since a non-customer will have to become a customer to order food. Therefore, after registering as a customer, they will see an additional tab with customer-specific functionalities. Lastly, the UI for the User interface should be very basic, with only one option to view all available food items.

4.2 Application Interface

The end-to-end user does not need to install any extra dependencies. The system requires the user to have a browser with JavaScript enabled, that is it! It is platform independent and can work on any operating system, such as Windows, MacOS, Linux, and Android. However, cryptography should be maintained throughout the system to ensure security, as the system can be accessed from anywhere via the internet.

4.3 Communication Interface

The communication interface between the server and client programs of ICEFoodswill utilize HTTP for transmitting hypertext. This means that any user with standard communication protocols will be able to communicate with ICEFoods without the need for protocol conversion or other difficulties.

4.4 User Management

User management is a crucial task for the admin, which involves inputting user IDs and approving new users.

|  |  |
| --- | --- |
| Numbers | Requirement |
| UM-001 | Create a new user account by providing the username, password, contact number, and email address. |
| UM-002 | Password should be given twice to match. |
| UM-003 | Username and password will be sent to the email address associated with the account. |
| UM-004 | Password should be in md5 format. |
| UM-005 | Approve a users’ information. |
| UM-006 | Freeze the editing option of users' information by the users. |

4.5Order Management

|  |  |
| --- | --- |
| Numbers | Requirements |
| OM-001 | Customer can order the food items. |
| OM-002 | Multiple items can be added to single cart. |
| OM-003 | Order can be modified by the specific customer. |
| OM-004 | An Order can be monitored by the specific customer and admin. |
| OM-005 | An Order can be canceled by the customer or admin. |

4.6Delivery Management

|  |  |
| --- | --- |
| Numbers | Requirements |
| DV-001 | Customer will pay money to Delivery guy when it will be cash on delivery. |
| DV-002 | Customer can also pay the money by various gateway like bkash, Roket, master card etc. |
| DV-003 | When the customer will receive the item then the customer will pay the price of order and delivery charge to delivery guy. |

4.8 Delivery Records

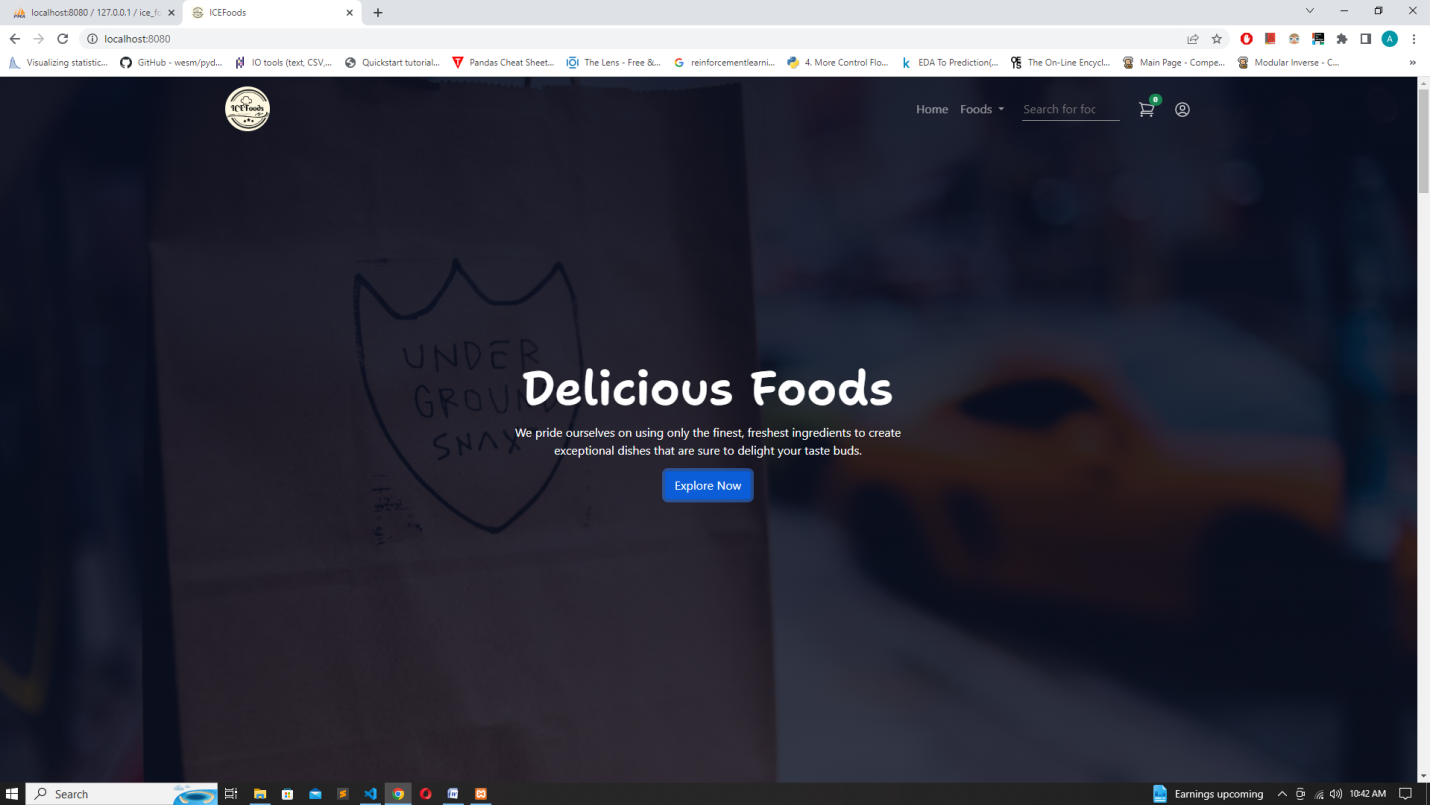
|  |  |
| --- | --- |
| Numbers | Requirements |
| PM-001 | Admin can check how many items has been delivered in a day. |
| PM-002 | Checking how many orders are pending today. |
| PM-003 | Controlling the orders delivery record. |

**5 Page and Form Design**

This section covers form and pages used into the system

**5.1 Home page**

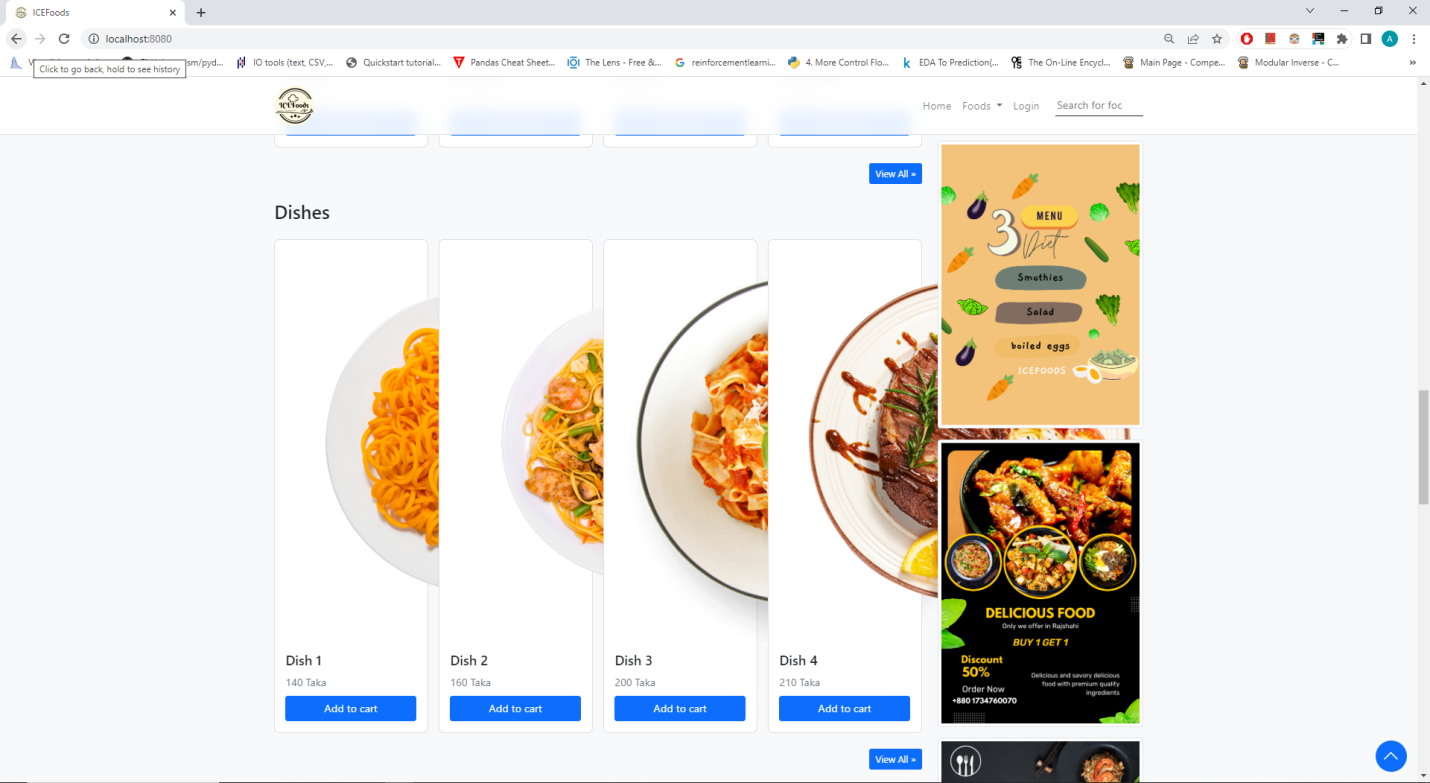
A home page or a start page is the initial or main web page of a website or a browser. The initial page of a website is sometimes called main page as well.

****

**5.2 Product Page**

Here all the products restaurant supplies will be displayedGraphical user interface, text, application, PowerPoint

Description automatically generated

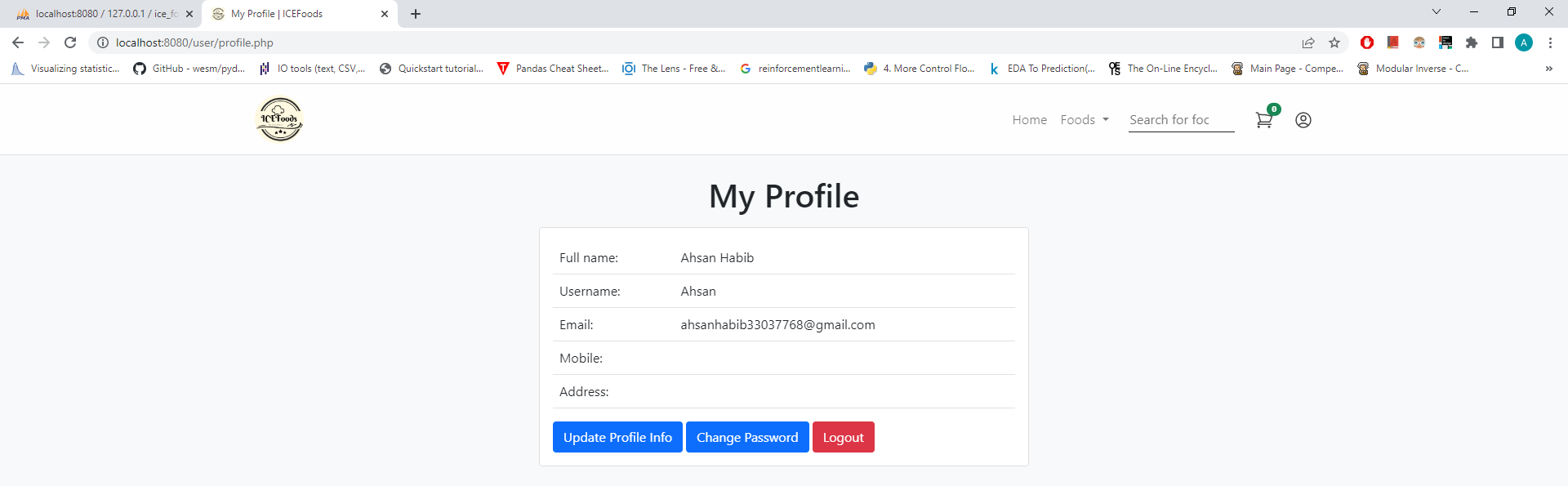


**5.3 Cart Page**

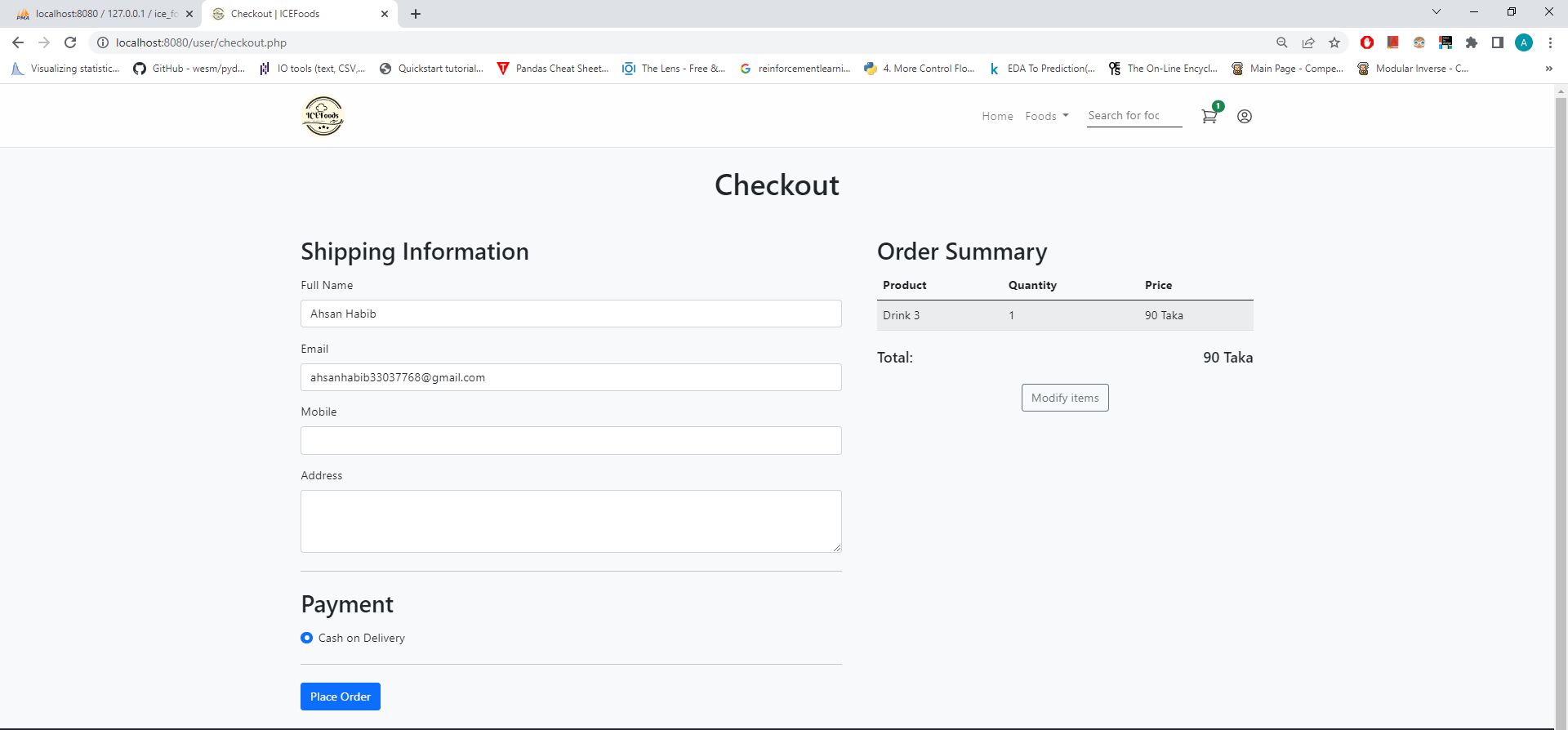
Graphical user interface

Description automatically generated

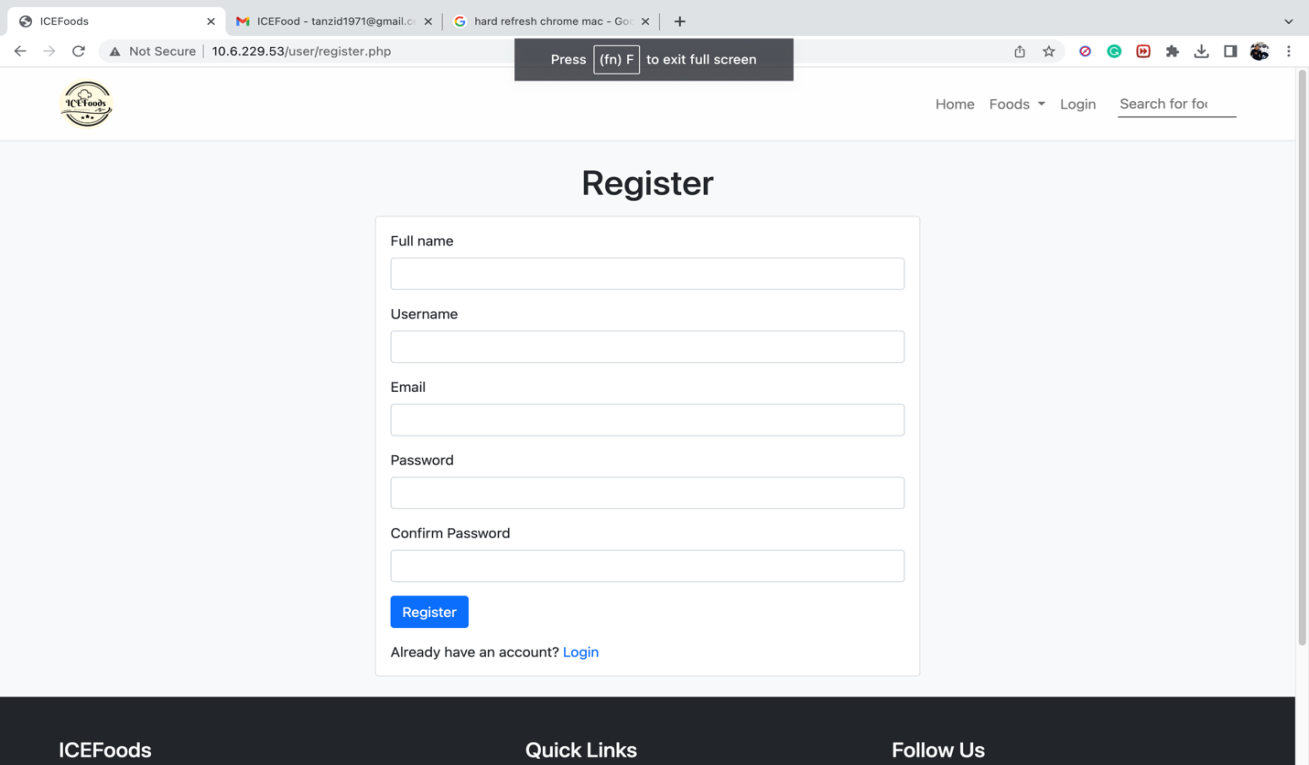
**5.4 Log in Page**



**5.5 Check out Page**

****

**5.6 User Registration Page**



6 RISK ORIENTATIONS

6.1 Overview

A risk assessment has conducted for ICEFoods.

**Technical Risks**

* System failure: The system may fail due to hardware or software issues, resulting in downtime and loss of revenue.
* Data breaches: The system may be vulnerable to cyber-attacks, resulting in the theft of customer and financial data.
* Integration issues: The system may not integrate effectively with other systems, resulting in data errors and delays.

**Operational Risks**

* Staff availability: The restaurant may not have enough staff to handle the volume of orders, resulting in delays and customer dissatisfaction.
* Equipment failure: The restaurant's equipment, such as ovens and refrigerators, may fail, resulting in delays and decreased quality of food.
* Delivery delays: The restaurant may face delays in delivering orders due to traffic, weather, or other unforeseen circumstances.

**Financial Risks**

* Payment processing issues: The system may experience delayed or failed payments, resulting in loss of revenue.
* Fraudulent activities: The system may be vulnerable to fraudulent activities, resulting in financial losses and damage to the restaurant's reputation.
* Incorrect pricing: The system may have incorrect pricing, resulting in overcharging or undercharging customers.

**Legal Risks**

* Data protection and privacy regulations: The system may not comply with data protection and privacy regulations, resulting in fines and legal action.
* Copyright violations: The system may include copyrighted material without permission, resulting in legal action.
* Other legal issues: The system may face legal issues related to taxes, contracts, and other legal obligations.

6.2 Mitigation Strategies

**Technical Risks**

* System failure: Implement a robust backup and recovery system to ensure that data is not lost and downtime is minimized.
* Data breaches: Implement appropriate security measures such as firewalls, encryption, and access controls to prevent data breaches.
* Integration issues: Conduct thorough testing and implement monitoring systems to identify and resolve integration issues.

**Operational Risks**

* Staff availability: Train and hire enough staff to handle the volume of orders and provide flexible schedules to ensure staff availability.
* Equipment failure: Conduct regular maintenance checks and have backup equipment available to minimize downtime.
* c. Delivery delays: Monitor and communicate with delivery personnel to anticipate and minimize delays.

**Financial Risks**

* Payment processing issues: Implement a secure payment gateway and conduct regular audits to ensure timely and accurate payments.
* Fraudulent activities: Implement fraud detection measures such as machine learning algorithms to identify and prevent fraudulent activities.
* Incorrect pricing: Implement a pricing validation process and monitor pricing regularly to ensure accuracy.

**Legal Risks**

* Data protection and privacy regulations: Comply with relevant data protection and privacy regulations such as GDPR and CCPA.
* Copyright violations: Ensure that all copyrighted material is properly licensed and obtain permission before using copyrighted material.
* Other legal issues: Conduct regular legal reviews and ensure compliance with all relevant legal obligations.

**Training**

* Provide training to staff on how to use the system effectively, including placing orders, managing orders, and using the reporting and analytics features.
* Provide training to staff on security best practices, including creating secure passwords, identifying phishing scams, and reporting suspicious activity.
* Provide training to staff on legal requirements, such as data protection and privacy regulations, copyright laws, and other legal issues relevant to the system.

**Review and Update**

* Conduct regular risk assessments to identify and evaluate potential risks.
* Update risk mitigation strategies regularly to reflect changes in the system and new potential risks.
* Review legal obligations regularly and ensure compliance with all relevant laws and regulations.

**7.1 Limitations**

* For upcoming improvement, there are some proposals to advance our project abilities.
* There is no email verification system.
* There is no online bill payment system.
* There is no security protection such as SSL, Site lock.
* SMS alert system is not available right now.
* There is no online secured payment getting system.

**7.2 Future Plans**

* We will add more features to improve our project.
* There will be email verification system.
* We will add SSL security system.
* New product update newsletter will be added.
* SMS alert system is easier for the customer.
* We also work on online payment gateway integration.
* Additionally, it is just a beginning. Supplementary the system may be used in various other types of reviewing process.

**8 Conclusion**

Online FOOD-Order management system for ICEFoods is for computerizing the working in a restaurant. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The restaurant managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average restaurant and is capable to provide easy and effective storage of information related to customers that come up to the restaurant. It provides the home delivery facilities to the customer. It also billing facility such as cash on or pay with bkash. The system is also provides location flexibilities in Rajshahi City.