**Design and Implementation of a Website of Online Restaurant Automation**



A project paper Submitted for the partial Fulfillment of the requirement for the degree of Bsc. Engineering of Department of Computer Science & Engineering.

**Project Supervised by**

Md. Tanvir Hossain

Lecturer

Department of Computer Science & Engineering

**Submitted By**

Roll: 13205445

Session: 2012-13

Registration Number: 2314

Course Code: CSE-4292

**Online Restaurant Automation**

**Acknowledgement**

At first I thank to almighty Allah for his kind blessing.

Then, I would like to express my gratitude and respect to my project supervisor, Md. Tanvir Hossain lecturer, Department of Computer

Science & Engineering, University of Rajshahi.

I am grateful for his inspiring encouragement, regular guidance, realistic appreciation and valuable suggestions throughout my project work. I am also thankful for his continuous feedback throughout this project work.

In completing this graduate project I have been fortunate to have help,support and encouragement from my friends.I would like to acknowledgement for their cooperation.

I would like to dedicate this project to my parents and my lovely friends.

The Author

**Abstract**

"Online restaurant automation" is a web based application where the ordering system is automated.

In my project there are three sections. The sections are admin,

customer and receptionist.Admin can add,edit,delete any item.Admin

can also set the next day menu.In customer section,there is dynamic

menu for item selection. Customer can add, edit, delete order.The selected

item of customer is detected by id oftable.In receptionist section, the

receptionist detect the table id,check any discount,calculate the bill and

print the bill.

CONTENTS

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Topics** | | | | **Page** | | | |
| **Chapter 1: Introduction** | | | | | | | | |
| 1.1 | Restaurant automation | | | | 8 | | | |
| 1.2 | Proposed system | | | | 8 | | | |
| 1.3 | Motivation | | | | 9 | | | |
| **Chapter 2: Project Objectives & Deliverables** | | | | | | | | |
| 2.1 | **Project Objectives & Deliverables** | | | | 11 | | | |
| **Chapter 3: Tools** | | | | | | | | |
| 3.1 | Scripting languages | | | | 13 | | | |
| 3.2 | HTML | | | | 14 | | | |
| 3.3 | CSS | | | | 15 | | | |
| 3.4 | Java script | | | | 16 | | | |
| 3.5 | Bootstrap framework | | | | 16 | | | |
| 3.6 | The reason for using Bootstrap in this project | | | | 17 | | | |
| 4.2 | Apache Web Server | | | | 17 | | | |
| 3.3 | The reason of using Laravel in my project | | | | 17 | | | |
| 3.4 | MySQL relational Database | | | | 18 | | | |
| 3.5 | The Reason of using MySQL in this project | | | | 18 | | | |
| **Chapter 4: System methodology** | | | | | | | | |
| 4.1 | System methodology | | | | 20-21 | | | |
| 4.2 | GANTT CHART | | | | 22 | | | |
| **Chapter 5: System analysis** | | | | | | | | |
| 5.1 | Features | | | | | 24 | | |
| **Chapter 6: System design** | | | | | | | | |
| 6.1 | | | | Use-case diagram | 26 | | | |
| 6.2 | | | | Database table | 27 | | | |
| 6.3 | | | | Data flow table | 28-30 | | | |
| **Chapter 7: System implementation** | | | | | | | | |
| 7.1 | Admin | | | | 32-33 | | | |
| 7.2 | Customer | | | | 34-36 | | | |
| 7.3 | Receiptionist | | | | 36 | | | |
| **Chapter 8: Evaluation and testing** | | | | | | | | |
| 8.1 | | Software testing | | | | | 38 | |
| 8.2 | | Content testing | | | | | 38 | |
| 8.3 | | interface testing | | | | | 39 | |
| 8.3 | | Navigation testing | | | | | 39 | |
| **Chapter 9: Discussion and conclusion** | | | | | | | | |
| 9.1 | | | Discussion | | | | | 41 |
| 9.2 | | | Future enhancement | | | | | 42 |
| 9.3 | | | Conclusion | | | | | 43 |
| **References** | | | | | | | | |

**Chapter [1]:Introduction**

**Introduction**

**Online Restaurant Automation:**

This document will propose all features and procedures to develop the system. These documents specially containing details about objectives, scope, design model, primary requirements and finally monitoring and reporting mechanisms.

It is a web application to automate the order system in a restaurant. In most cases in the restaurant we get printed menu to choice food item.

But here we faces some problems such as sometimes the waiter says that this item is not available today. Again manager faces problems to

Set the menu for the next day. Cashier faces some problems in preparing cash memo.If we want to take review from the customer this is not so easy. To solve these kinds of problem I am proposing system.

Here customers can order food items from a dynamic menu, managers can set menu for the next day dynamically and this system makes easy to

Provide cash memo to the customer.Finally here we can take review from the customer.

My proposed system provides these:

* Dynamic food choice
* Can order as many as wish
* Fast order system.
* We are using Tablet Pc to take order from the customers to make it easy to select the food items
* Admin will set the menu previous night, which will be available the next day.

My motivation is to build this system are these :

* To make customer's menu selection easier and dynamic
* To give flexibility of setting menu of next day
* To make the payment easier

**Chapter [2]:Project Objectives & Deliverables**

The purpose of the project is to design and implement a web application where order system is automated.

In Rajshahi, there is no automation system in any restaurant. Sometimes Customer order item that is not available that day.So this is not a good side for any restaurant.So my system is obviously a fast order system. The project includes a complete level of development with documentation.

Project Deliverables:

* Project Plan
* system Design Document
* Test Plan
* Test Report
* Final Report
* Product
* User Manual

**Chapter [3]:Tools**

This chapter discusses the background study needed to develop the project.

For the purpose of the project, detailed study is done on the following topic...

* HTML
* CSS
* Bootstrap Framework
* JavaScript
* PHP
* Laravel Framework 5.1

**Scripting Language**

Scripting languages are the basic of much the CGI-BIN programming that is currently used to add a limited form of interactivity to web pages. We can write simple scripts to add an alert box or a bit of text to our pages or more complicated scripts that load particular page according to our visitor browser or change a frame back ground color depending on where they point the mouse. Scripting languages are little bit different from other programming language. They may be compiled / interpreted but they remain in some plain text or with the HTML codes that everybody can see the code and modify the code if necessary.

Again, scripting language can be categorized into two main type

* Server-side Scripting language
* Client-side Scripting language

Some examples of scripting languages:

Server-side Scripting language:

* PHP (Hypertext Processor)
* ASP (Active Server Page)
* JSP (Java Server Page)

Client side scripting language:

* VBScript
* Java Script etc

**HTML (Hypertext Markup Language):**

**Hypertext Markup Language** (**HTML**) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag*. In between these tags web designers can add text, further tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

**CSS (Cascading Style Sheet):**

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. It’s most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

**JavaScript:**

JavaScript is a high-level, [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [untyped](https://en.wikipedia.org/wiki/Programming_language), and [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) programming language. It has been standardized in the [ECMAScript](https://en.wikipedia.org/wiki/ECMAScript) language specification. Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), it is one of the three core technologies of Web content production; the majority of [websites](https://en.wikipedia.org/wiki/Website) employ it and it is supported by all modern [Web browsers](https://en.wikipedia.org/wiki/Web_browser) without [plug-ins](https://en.wikipedia.org/wiki/Browser_extension). JavaScript is [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) with [first-class functions](https://en.wikipedia.org/wiki/First-class_function), making it a [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm) language, supporting [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming),[[8]](https://en.wikipedia.org/wiki/JavaScript) [imperative](https://en.wikipedia.org/wiki/Imperative_programming), and [functional](https://en.wikipedia.org/wiki/Functional_programming) programming styles.[[6]](https://en.wikipedia.org/wiki/JavaScript) It has an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for working with text, [arrays](https://en.wikipedia.org/wiki/Array_data_type), dates and [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), but does not include any [I/O](https://en.wikipedia.org/wiki/Input/output), such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Although there are strong outward similarities between JavaScript and Java, including language name, [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)), and respective [standard libraries](https://en.wikipedia.org/wiki/Standard_library), the two are distinct languages and differ greatly in their design. JavaScript was influenced by programming languages such as [self](https://en.wikipedia.org/wiki/Self_(programming_language)) and [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)).

JavaScript is also used in environments that are not Web-based, such as [PDF](https://en.wikipedia.org/wiki/Portable_Document_Format) documents, [site-specific browsers](https://en.wikipedia.org/wiki/Site-specific_browser), and [desktop widgets](https://en.wikipedia.org/wiki/Desktop_widget). Newer and faster JavaScript [virtual machines](https://en.wikipedia.org/wiki/Virtual_machine) (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side [Web applications](https://en.wikipedia.org/wiki/Web_application). On the client side, JavaScript has been traditionally implemented as an [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)) language, but more recent browsers perform [just-in-time compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation). It is also used in game development, the creation of desktop and mobile applications, and server-side network programming with run-time environments such as [Node.js](https://en.wikipedia.org/wiki/Node.js).

**Bootstrap Framework:**

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optical JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap originally named Twitter Blueprint, was developed by mark Otto and Jacod Thornton at Twitter as a framework to encourage consistency across internal tools. It provides are Style sheets, Re-usable component, Java Script and so on.

**The reason for using Bootstrap in this project**

1. Easy to get started.
2. Great grid system.
3. Base styling for most HTML elements
4. Extensive list of components.
5. Bundled JavaScript plugins.

**Apache Web Server:**

Apache is the most widely used HTTP server in the world today. It surpasses all free and commercial competitors on the market and provides a myriad of features. It is also the most widely used web server for a Linux system. A web server like apache in its simplest function , is software that displays and serves HTML pages hosted on a server to a client browser that understand the HTML code. Mixed with third party modules and programs, it can become powerful software, which will provide strong and useful services to a client browser.

**Laravel 5.1:**

**Laravel** is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar.

**The reason of using Laravel in my project**

* Excellent Documentation.
* Laracasts
* Intuitive Syntax
* Practical Application Structure
* Artisan Code Generation
* Out-of-the-box User Model
* Blade Templating Engine
* Dependency Injection Made Simple
* Supporting Products and Packages
* Innovative Founder

**MySQL relational Database**

MySQL is a freely available RDBMS (Relational Database Management System).It is a true multi-user .multi-threaded SQL database server. SQL(Standard Query Language) is the most popular and standardized database language of the world. MySQL is a client/server implementation that consists of server daemon MYSQL and many different client programs and libraries.

**The Reason of using MySQL in this project:**

* Scalability and Flexibility
* High Performance
* High Availability
* Robust Transactional Support
* Web and Data Warehouse Strengths
* Strong Data Protection
* Comprehensive Application Development
* Management Ease
* Open Source Freedom and 24 x 7 Support
* Lowest Total Cost of Ownership

**Chapter [4]**

**System methodology**

**System methodology**

The waterfall model is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are −

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

**GANTT CHART**

**Chapter [5]: System Analysis**

**Features:**

There are three main sections or features.

The features are given below:

**Admin:**

* Set manu item everyday for the next day.
* Can add,delete , edit food item.(item details)

**Customer:**

* Traverse menu
* Order food
* Can add or cancel order
* Confirm order(everything will be done by taking table number as customer id)

**Receptionist:**

* Detect the order from table number.
* Check any discount.
* Print the bill.

**Chapter [6]: System Design**

**Use- case diagram**

**Add/edit/delete item**

AA A

AA

**set menu of the next day**

**Order food**

**Edit order**

**Confirm order**

**View order**

**Print bill**

**Database table:**

**Users:**

|  |  |  |
| --- | --- | --- |
| **Fields** | **Type** | **Size** |
| Id(primary key) | int | 11 |
| name | varchar | 191 |
| email | varchar | 191 |
| password | varchar | 191 |

**Foods:**

|  |  |  |
| --- | --- | --- |
| **Fields** | **Type** | **Size** |
| Id(primary key) | int | 11 |
| name | varchar | 191 |
| category | varchar | 191 |
| status | varchar | 191 |
| price | varchar | 191 |
| Minimum\_time | varchar | 191 |
| photo | varchar | 191 |
| available | int | 11 |

**Receiptionist:**

|  |  |  |
| --- | --- | --- |
| **Fields** | **Type** | **Size** |
| Id(primary key) | int | 11 |
| Table\_number | varchar | 191 |
| Item\_number | varchar | 191 |
| Item\_quantity | varchar | 191 |

**Data flow diagram:**

**Add item**

**Edit/delete item**

**Set the next day menu**

**Customer:**

**Order food**

**Edit food**

**Confirm order**

**Receiptionist:**

**View order**

**Print bill**

**Chapter [7]**

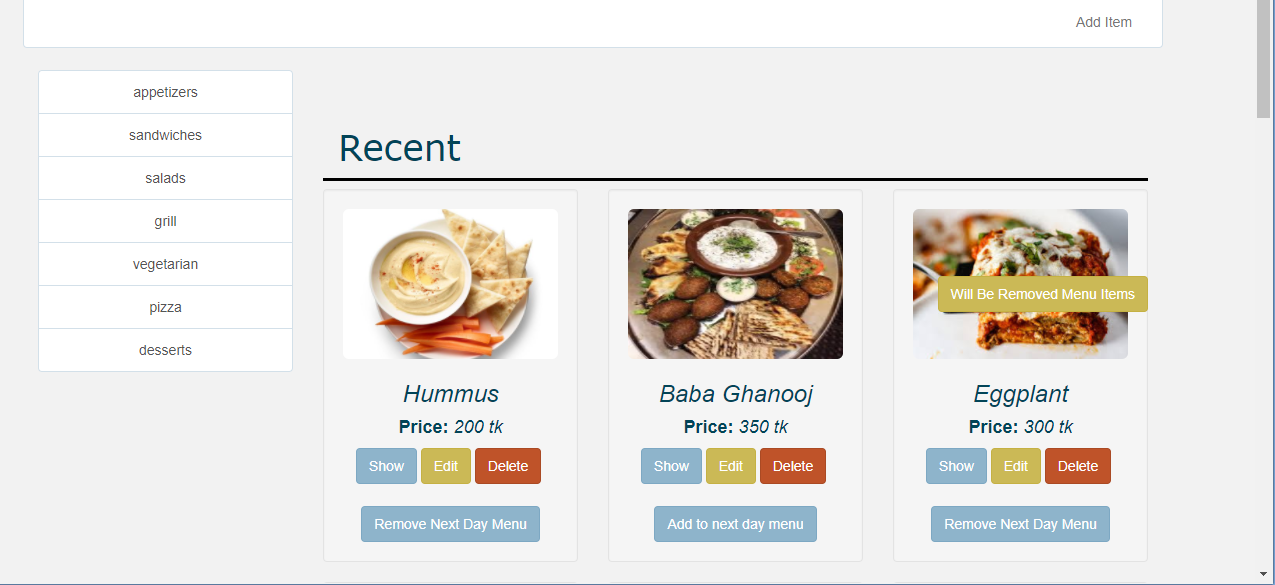
**System**

**Implementation**

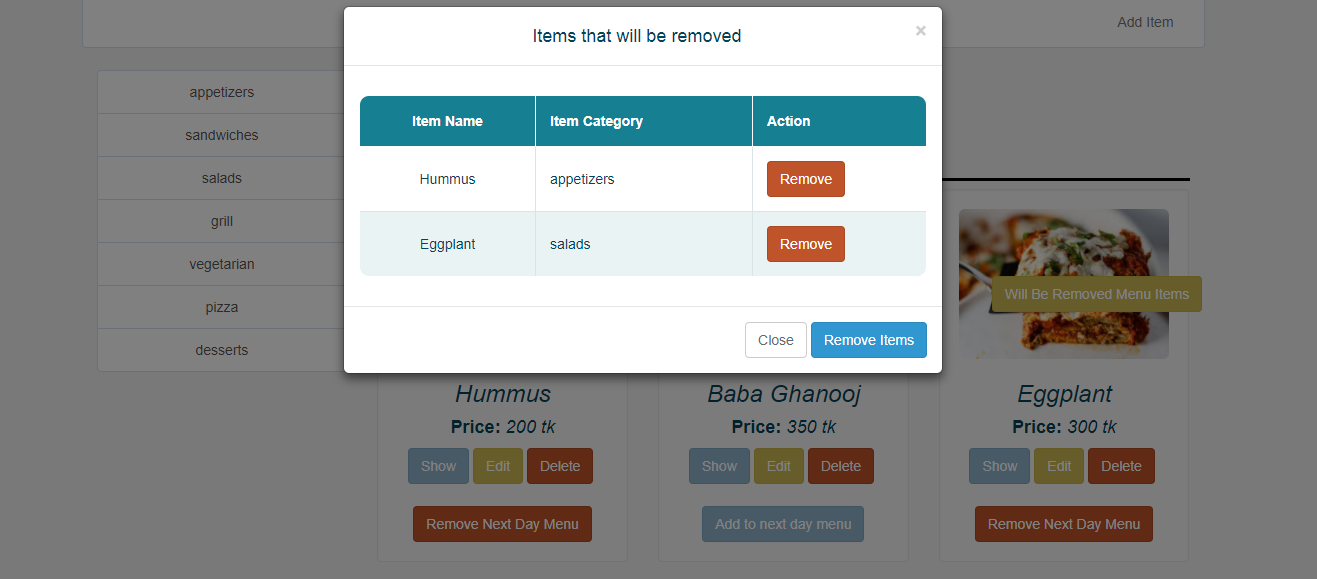
There are three sections in my project.

* Admin
* Customer
* Receptionist

**Admin:**

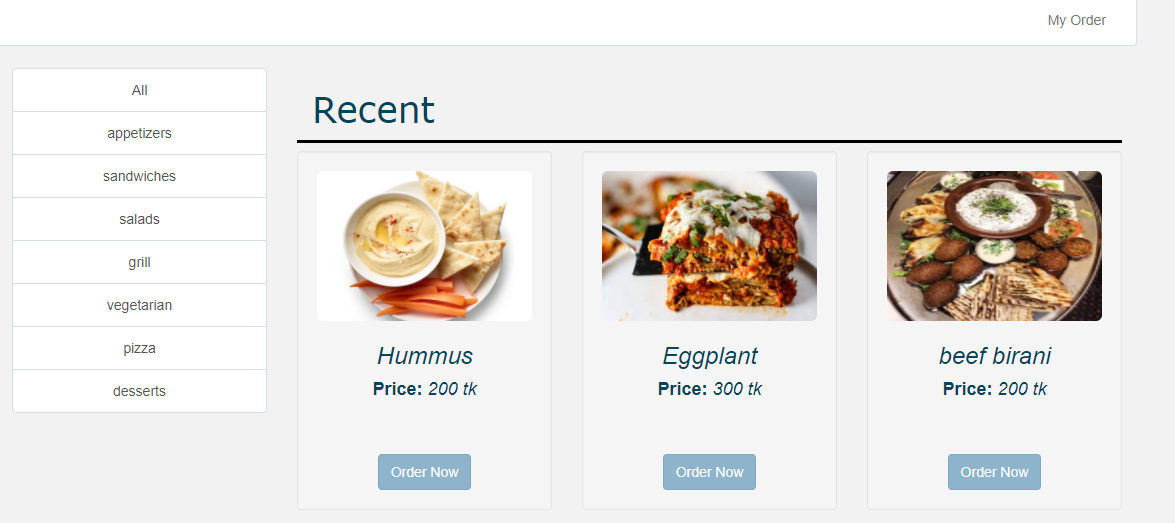


Admin can select item to remove for the next day by clicking remove next day menu or can add item by clicking add the next day menu.By clicking remove the next day menu and will be removed menu items we will see the following window:

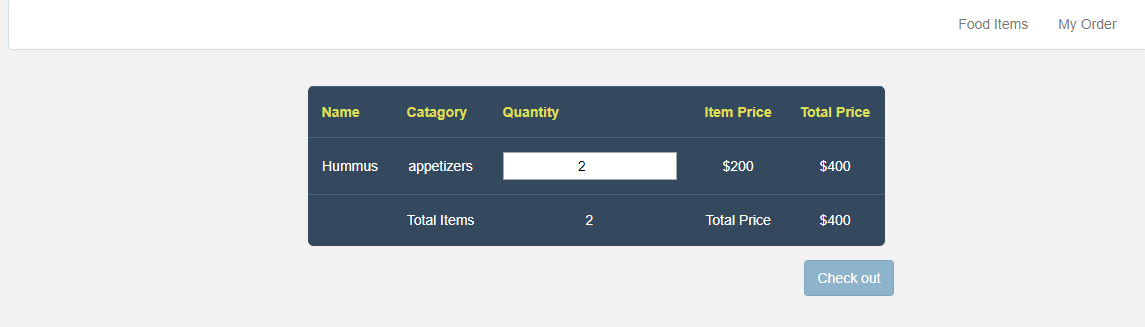


**Customer:**

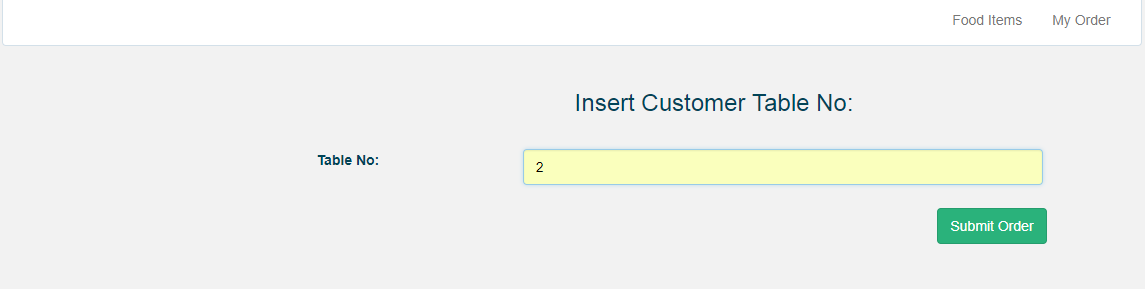
The dynamic menu available for customer.The customer can order any item.



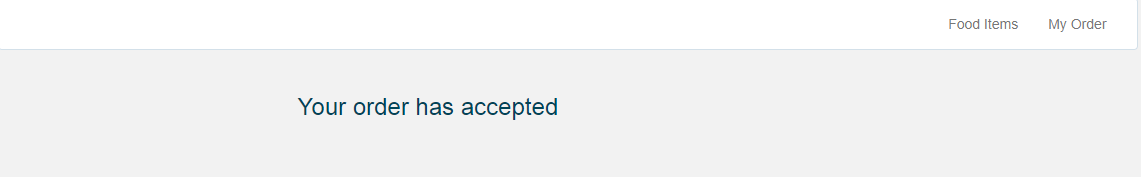
By clicking order now customer can see the window that will show the ordered food,category,quantity,price.



By clicking check out button customer will see insert the table number.

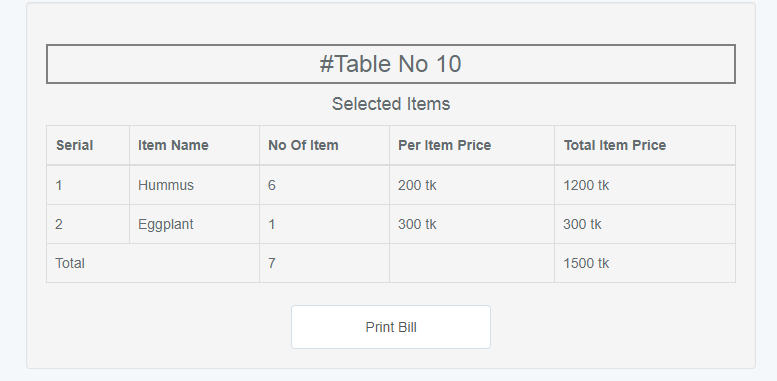


By clicking the submit order button,customer will see the order is accepted.



**Receptionist:**

Receptionist can view order and print the bill.



**Chapter [8]**

**Evaluation and Testing**

***Software testing*:**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understands the risk of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Computer program/application/product:

* Meets the requirements that guided its design and development,
* Works as expected,
* Can be implemented with the same characteristics,
* And satisfies the needs of stakeholders.
* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has a probability of finding an as yet undiscovered error
* A successful test is one that uncovers an undiscovered error.

***Content testing:***

* Attempts to uncover errors in the content that is present as a part of virtually every Web App.
* This testing activity is similar in many respects to copy editing for a writing document.
* In fact, the development time for a large Web Apps might include (or content) the services of a professional copy editor or uncover typographical errors, grammatical mistakes, errors in content consistency, error in graphical representations and cross-referencing errors.
* In addition to examining static content for errors, this testing step also considers dynamic content derived from data maintained as part of a database system that has been integrated with the Web App. In many cases this form of testing many also need to continue on the ongoing basis as new content is added during the life of the Web App.

***Interface testing:***

* Exercises interaction mechanisms and validates aesthetic aspects of the user interface.
* The intent is to uncover errors that result from poorly implemented interaction mechanisms or omissions, inconsistencies or ambiguities that have been introduced into the interface inadvertently.

***Navigation testing:***

* Navigation links, these mechanisms include internal links within the Web App, external links to other Web Apps and anchors within a specific web page.
* Redirect, these links come into play when a user requests a nonexistent URL or select a link whose destination has been removed or whose name has changed.
* Bookmarks, although bookmarks are a browser function, the Web App should be tested to ensure that a meaningful page title can be extracted as the bookmark is created and the dynamic pages are bookmarked appropriately.
* Fames and framesets, each frame contains the content of a specific web pages, a frameset contains multiple frames and enables the display of multiple web pages at the same time
* Site maps, a site map provides a complete table of contents for all web pages
* Internal search engine, an internal (local) search engine allows the user to perform a key word search within the Web App to find needed content.

**Chapter [9]**

**Discussion and conclusion**

**Discussion:**

In Rajshahi, there is no automation system in any restaurant i saw.This encourages to build this project. Customer cannot get their desired item. Here we faces some problems such as sometimes the waiter says that this item is not available today. Again manager faces problems to set the menu for the next day. Cashier faces some problems in preparing cash memo.I hope this project will remove this kind of problems.It will really helpful for all users.

**Future enhancement:**

This is my initial approach to a web-based Algorithm online restaurant automation system. Further enhancement of this system will be developed with various enhancement features. This project is not fully completed because of lack of time. So, this project future plan is to develop the unfulfilled requirements.

Scope of enhancement:

* Customer review system where customer can get their opinion about food.
* Improve view more.
* Improve user interface.

**Conclusion:**

There is always a scope of betterment and this system is not against this perception. This project is especially designed for reducing time for customer to choose their item and managing the ordering system efficiently and also to improve the working standard through computerization.Because of lack of time there are a few limitations and these can be removed in the future.

**References:**

* Software Engineering by Sommervile
* System Analysis and Design – by Elias M. Award
* Database System Concepts –by Silberschatz, Korth & Sudarshan
* <https://en.wikipedia.org/wiki/HTML>
* <https://en.wikipedia.org/wiki/Cascading_Style_Sheets>
* <https://en.wikipedia.org/wiki/JavaScript>
* https://laravel.com