**CANTEEN AUTOMATION**

**ABSTRACT**

The project “Canteen Automation” is implemented to reduce the manual work and enhances the accuracy of work in a restaurant. This system manages and maintains the record of customers and room in the hotel. The room has different categories such as A/c., non-A/c., dormitory etc., so the charges and records will be maintained accordingly. This software has been made in a user friendly interface. So that normal person can add and delete the entries of customers and handle all the transactions easily. This project is also designed with full consideration to help the users in an easy manner without any unnecessary wastage of time.

This system can be implemented in big restaurant where customers can order their food from their room using system. The menu card consists of various food varieties available in the restaurant. Through the ordering form, the customer can simply click and order the food. The messaging system tells the supplier to supply the particular food. The billing system prepares the bill according to the delivered food. This system entirely reduces the unnecessary time waste inside the hotel as well as it reduces unnecessary noise.

**1. INTRODUCTION**

In this project admin has enter login and created menu list for customers, then these menu list are shown in the customer menu list. customer will view the menu list and order their favorite food, then the order food was send to the pending orders , after the food preparing the server will deliver the food items. The project is aimed to develop by **JAVA** as Front end and **MS SQL SERVER** as Back end. The back end is used to store the information in this system.

**1.1 SYSTEM SPECIFICATION**

**1.1.1 HARDWARE SPECFICATION:**

* Processor : P 4 700 GHz.
* RAM : 4GB RAM
* Hard Disk Drive : 40 GB HDD

**1.1.2 SOFTWARE SPECIFICATION:**

* + Operating System : Windows XP/7/8/10
  + Front End : JAVA
  + Back End : MY SQL

1. **SYSTEM STUDY**

**2.1 EXISTING SYSTEM:**

Before the canteen automation system a customer need to first ask menu list from waiter then choose menu then again call waiter then order the food. Which could be takes more time to ordering a single food. Sometimes if the waiter has busy with some customer it make some irritate feeling to the canteen. So its very hard to maintain the canteen system.

**2.1.1 DRAWBACKS:**

The existing system has the following drawbacks.

* Customer don’t know about the food details
* Waste of time to waiting for a waiter.
* Easy to ordering a food

**2.2 PROPOSED SYSTEM:**

This system helps to ordering a food manually by the customer. Customer can choose their favourite food without asking the waiter. This might be reducing time of ordering a food. Every tables have one system, with this system helps to ordering a food, then the order will be visible to the cooking team they will saw and preparing and delivering a food.

**2.2.1 FEATURES:**

* Attractive user interface
* Every user can be easily to use this software.
* Taking less time to ordering a food

1. **SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

**3.2 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
* will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.3 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.4 DATABASE DESIGN**

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

### Business needs addressed:

* Determine the basic objects about which the information is stored
* Determine the relationships between these groups of information and the objects
* Effectively manage data and create intelligent information
* Remote database administration or on site administrative support
* Database creation, management, and maintenance
* Information retrieval efficiency, remove data redundancy and ensure data security

**3.5 SYSTEM DEVELOPMENT**

**3.5.1 DESCRIPTION OF MODULES**

1. Waiter login
2. Menu Items
3. Order Menu items
4. Delivery Item

**Waiter Login**

A waiter can entered the login and view his orders. He can view the order and accepted. Once the waiter accept the order the message will be goes to the customer.

**Menu Items**

Menu items was added by admin, these menu items will change daily base. Customer can easily find what are the dishes are available now, then they will order his favourite food.

**Order Menu Items**

Once customer was give an order the order was send to the waiter, the waiter will be verify and delivered that order.

**Delivery Item**

After the food was prepared the notification will be send to the customer and waiter, then the waiter can pick the order and delivered.

1. **SYSTEM TESTING AND IMPLEMENTATION**

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TESTING METHODOLOGIES**

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfills this objective a series of test step unit, integration. validation and system tests were planned and executed the test steps are:

**System Testing**

Testing is an important phase in project development. System testing makes a logical assumption that if all parts of the system are correct, and the goal will be achieved successfully. The software must meet the user specification and it must satisfy according to the needs of the users.

Testing is the process of executing a project within the intend of finding errors. A good test case is one that has a high probability of finding an undiscovered error.

**Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

**In Project**, Each module such menu items registration module, customer module, food details details module, orders module are tested individually for example, Customer details module can contain the more forms to maintain the information so all forms could be tested like entered information store appropriately in database access page or not. If correctly accessed means the testing of registration module successfully completed. Likewise all modules are tested successfully.

**Integration Testing**

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

**In Project,** Integration of two modules can be tested together such as menu items registration details and admin login module for verification purposes providing proper accessibility to users. The communication of Registration and Login module can test and executed successfully.

**Validation Testing**

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

**In this project,** Admin login details form Enter without username and password in textbox enter the submit button then Login failed message otherwise checks the both textbox value that is true means valid page displayed. Enter Password Displaying password character \*.if it displays the characters security is not availed so testing of software is failed.

**Output Testing**

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**In project** All the forms are tested as it gives the necessary output to the user’s search such as view response details.

1. **CONCLUSION:**

This project main goal is that we no need to waiting for waiter for booking our favorite food. Which is fully automatically to go through online and order our foods. Our main aim of this project is that configure automatically.

**FUTURE ENHANCEMENT**

In our future enhancement this application has an billing from using this application, the billing modules was maintained by the manager, so it will keep safe and secure. then the screen design was fully changed it's an attractive user friendly design. another one process customer can book their food in home.

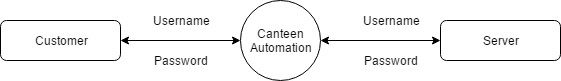
**BIBLIOGRAPHY**

* **CH97**  
  Gary Cornell and Cay S. Horstmann, Core Java, second ed., SunSoft Press, 1997
* **ELW98**  
  Robert Eckstein and Marc Loy and Dave Wood, Java Swing, O'Reilly, 1998.
* **Flanagan96**  
  David Flanagan, Java in a Nutshell, second ed., O'Reilly, 1996.
* **Gea99b**  
  David M. Geary Graphic Java 2: Mastering the JFC, vol. II, Swing, third ed., Sun Microsystems Press, 1999.

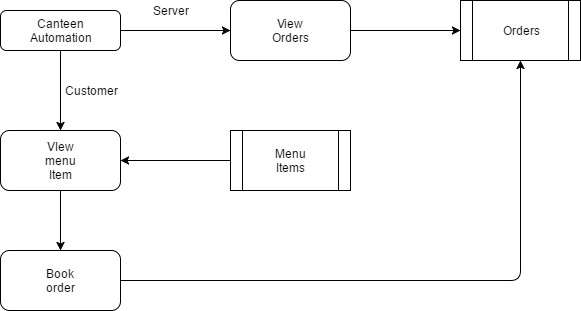
**APPENDICES**

1. **DATA FLOW DIAGRAM**

LEVEL 0:

****

LEVEL 1:

****

1. **TABLE STRUCTURE**

**TABLE NAME : ADMIN**

**PRIMARY\_KEY : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int |  | Admin id |
| Username | Varchar | 30 | Admin username |
| password | Varchar | 30 | Admin password |

**TABLE NAME : MENU**

**PRIMARY KEY : MENUID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Menuid | Int | 10 | Menu id |
| Name | Varchar | 30 | Menu name |
| Amount | Int | 10 | Amount |
| Img | Varchar | 30 | Image path |

**TABLE NAME : ORDERS**

**PRIMARY KEY : id**

**FOREIGN KEY : menu\_id,tid**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int | 10 | Order id |
| Menu\_id | Int | 10 | Menu\_id |
| Tid | Int | 10 | Table id |
| Quantity | Int | 10 | auantity |

1. **SAMPLE CODING**

|  |
| --- |
| <!DOCTYPE html> |
|  |

|  |
| --- |
| <html lang="en"> |
|  |

|  |
| --- |
| <head> |
|  |

|  |
| --- |
| <meta charset="UTF-8"> |
|  |

|  |
| --- |
| <link href="/bootstrap/bootstrap.css" rel="stylesheet"/> |
|  |

|  |
| --- |
| <link href="/home/login.css" rel="stylesheet"/> |
|  |

|  |
| --- |
| <title></title> |
|  |

|  |
| --- |
| </head> |
|  |

|  |
| --- |
| <body> |
|  |

|  |
| --- |
| <div class="simple-login-container"> |
|  |

|  |
| --- |
| <h2>Canteen Automation System</h2> |
|  |

|  |
| --- |
| <div class="row"> |
|  |

|  |
| --- |
| <div class="col-md-12 form-group"> |
|  |

|  |
| --- |
| <input id="username" type="text" class="form-control" placeholder="Username"> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div class="row"> |
|  |

|  |
| --- |
| <div class="col-md-12 form-group"> |
|  |

|  |
| --- |
| <input id="password" type="password" placeholder="Enter your Password" class="form-control"> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div class="row"> |
|  |

|  |
| --- |
| <div class="col-md-12 form-group"> |
|  |

|  |
| --- |
| <input id="login" type="button" class="btn btn-block btn-login" value="Login " > |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <script src="/bootstrap/jquery.min.js"></script> |
|  |

|  |
| --- |
| <script src="/home/login.js"></script> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </body> |
|  |

</html>

|  |
| --- |
| <!DOCTYPE html> |
|  |

|  |
| --- |
| <html lang="en"> |
|  |

|  |
| --- |
| <head> |
|  |

|  |
| --- |
| <meta charset="UTF-8"> |
|  |

|  |
| --- |
| <link href="/bootstrap/bootstrap.css" rel="stylesheet"/> |
|  |

|  |
| --- |
| <link href="/home/user.css" rel="stylesheet"/> |
|  |

|  |
| --- |
| <title></title> |
|  |

|  |
| --- |
| </head> |
|  |

|  |
| --- |
| <body> |
|  |

|  |
| --- |
| <div class="navbar-wrapper"> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <div class="row"> |
|  |

|  |
| --- |
| <div class="col-sm-7"> |
|  |

|  |
| --- |
| <h1>Menu Orders</h1> |
|  |

|  |
| --- |
|  |
|  |

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

|  |
| --- |
| <div class="row" style="margin-top:20px;" id="menu\_body"> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div class="col-sm-4"> |
|  |

|  |
| --- |
| <h1>Your Cart</h1> |
|  |

|  |
| --- |
| <table class="table table-bordered"> |
|  |

|  |
| --- |
| <thead> |
|  |

|  |
| --- |
| <tr> |
|  |

|  |
| --- |
| <th scope="col">#</th> |
|  |

|  |
| --- |
| <th scope="col">Menu</th> |
|  |

|  |
| --- |
| <th scope="col">Price</th> |
|  |

|  |
| --- |
| </tr> |
|  |

|  |
| --- |
| </thead> |
|  |

|  |
| --- |
| <tbody id="cart\_body"> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </tbody> |
|  |

|  |
| --- |
| </table> |
|  |

|  |
| --- |
| <div style="float: right"> |
|  |

|  |
| --- |
| <button id="order" type="button" class="btn btn-warning">Order Now</button> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <script src="/bootstrap/jquery.min.js"></script> |
|  |

|  |
| --- |
| <script src="/home/usermenu.js" ></script> |
|  |

|  |
| --- |
| </body> |
|  |

|  |
| --- |
|  |
|  |

</html>

|  |
| --- |
| <html> |
|  |

|  |
| --- |
| <head> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <link href="/bootstrap/bootstrap.css" rel="stylesheet"/> |
|  |

|  |
| --- |
| <link href="/bootstrap/datatables.css" rel="stylesheet"/> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <!--<link rel="stylesheet" type="text/css" href="/home/home.css">--> |
|  |

|  |
| --- |
| </head> |
|  |

|  |
| --- |
| <body> |
|  |

|  |
| --- |
| <div class="container"> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <div class="row"> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <nav class="navbar navbar-default"> |
|  |

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

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

|  |
| --- |
| <a class="navbar-brand" href="#">Canteen Automation</a> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <ul class="nav navbar-nav"> |
|  |

|  |
| --- |
| <li><a href="#htab1" data-toggle="tab">Add Menu</a></li> |
|  |

|  |
| --- |
| <li><a href="#htab2" data-toggle="tab">View Orders</a></li> |
|  |

|  |
| --- |
| <li><a href="#htab3" data-toggle="tab">Completed orders</a></li> |
|  |

|  |
| --- |
| <li><a href="/login" >Logout</a></li> |
|  |

|  |
| --- |
| </ul> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </nav> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <div class="tab-content" style="margin: 30px"> |
|  |

|  |
| --- |
| <div role="tabpanel" class="tab-pane fade in active" id="htab1"> |
|  |

|  |
| --- |
| <h1>Add Menu</h1> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <div class="col-md-3"> |
|  |

|  |
| --- |
| <div class="form-group"> |
|  |

|  |
| --- |
| <label>Menu Name:</label> |
|  |

|  |
| --- |
| <input id="menuname" type="text" class="form-control"> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <div class="form-group"> |
|  |

|  |
| --- |
| <label>Amount:</label> |
|  |

|  |
| --- |
| <input id="amount" type="text" class="form-control"> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <button id="addmenu" type="button" class="btn btn-warning">Add Menu</button> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div class="col-md-3"> |
|  |

|  |
| --- |
| <table class="table table-bordered"> |
|  |

|  |
| --- |
| <thead> |
|  |

|  |
| --- |
| <tr> |
|  |

|  |
| --- |
| <th scope="col">#</th> |
|  |

|  |
| --- |
| <th scope="col">Menu Name</th> |
|  |

|  |
| --- |
| <th scope="col">Amount</th> |
|  |

|  |
| --- |
| <!-- <th scope="col">Action</th> --> |
|  |

|  |
| --- |
| </tr> |
|  |

|  |
| --- |
| </thead> |
|  |

|  |
| --- |
| <tbody id="menu\_body"> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </tbody> |
|  |

|  |
| --- |
| </table> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div role="tabpanel" class="tab-pane fade" id="htab2"> |
|  |

|  |
| --- |
| <h1>View Orders</h1> |
|  |

|  |
| --- |
| <div class="col-md-9"> |
|  |

|  |
| --- |
| <table class="table table-bordered"> |
|  |

|  |
| --- |
| <thead> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <tr> |
|  |

|  |
| --- |
| <th scope="col">#</th> |
|  |

|  |
| --- |
| <th scope="col">Table</th> |
|  |

|  |
| --- |
| <th scope="col">Menu Name</th> |
|  |

|  |
| --- |
| <th scope="col">Amount</th> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </tr> |
|  |

|  |
| --- |
| </thead> |
|  |

|  |
| --- |
| <tbody id="pending\_body"> |
|  |

|  |
| --- |
| </tbody> |
|  |

|  |
| --- |
| </table> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div role="tabpanel" class="tab-pane fade in" id="htab3"> |
|  |

|  |
| --- |
| <h1>Completed orders</h1> |
|  |

|  |
| --- |
| <div class="col-md-8"> |
|  |

|  |
| --- |
| <table class="table table-bordered"> |
|  |

|  |
| --- |
| <thead> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| <tr> |
|  |

|  |
| --- |
| <th scope="col">#</th> |
|  |

|  |
| --- |
| <th scope="col">Table</th> |
|  |

|  |
| --- |
| <th scope="col">Menu Name</th> |
|  |

|  |
| --- |
| <th scope="col">Amount</th> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </tr> |
|  |

|  |
| --- |
| </thead> |
|  |

|  |
| --- |
| <tbody id="completed\_body"> |
|  |

|  |
| --- |
| </tbody> |
|  |

|  |
| --- |
| </table> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| <div role="tabpanel" class="tab-pane fade in" id="htab4"> |
|  |

|  |
| --- |
| <h1>tab 4</h1> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </div> |
|  |

|  |
| --- |
| </body> |
|  |

|  |
| --- |
| <script src="/bootstrap/jquery.min.js"></script> |
|  |

|  |
| --- |
| <script src="/home/home.js"></script> |
|  |

|  |
| --- |
| <script src="/bootstrap/datatable.min.js"></script> |
|  |

|  |
| --- |
| <script src="/bootstrap/datatable.bootstrap.min.js"></script> |
|  |

|  |
| --- |
| <script src="/bootstrap/bootstrap.min.js"></script> |
|  |

|  |
| --- |
|  |
|  |

</html>