



Yamifood system guide.

By: Php framework=> Laravel 7





YAMIFOOD Restaurant

منفذين المشروع :

11868143	عبد الرحمن على خلف احمد
11868383	محمد معروف على
11868462	محمود يعقوب رمضان احمد
11868833	سلمى عادل

تحت اشراف :
دكتور / عاطف رسلان
معيد / جوزيف عبد الله

اهداءات



مضت الكثير من السنوات منذ دخولنا الكلية ، وفي كل يوم نتعلم اشياء جديدة يوما بعد يوم ،منذ بدء اليوم الاول ونحن لدينا الكثير من الاهداف نحققها وبفضل الله اولا ثم بفضل كل من ساندنا من اولياء الامر ومن السادة الافاضل اعضاء هيئة التدريس بارك الله فيكم ، قمنا بتحقيق الكثير والكثير من الاهداف على مدار السنين الماضية وها نحن هنا الان نصل الى تحقيق هدف من الاهداف العظمى التي اردنا تحقيقها فى المكان العظيم.

ونخص بالشكر هؤلاء السادة اعضاء هيئة التدريس:

أ / د : وجدى لويس



أ / د : سيد مشعال (عميد المعهد)



د : عاطف رسنان



أ / د : مدحت لويس



محتويات الكتاب

الباب الاول : الهيكل العام للنظام (20 : 9)

الصفحة	الموضوع
9	1- مقدمة عن المشروع بصفة عامة
12 : 10	2- شرح عمل النظام
13	3- المتطلبات الرئيسية للمشروع
14	4- اهداف المشروع
20 : 15	5- اللغات والبرامج المستخدمة

الباب الثانى : تحليل النظام (28 : 21)

رسومات قاعدة البيانات	
21	<i>Entity relationship diagram</i>
22	<i>Relational schema</i>
رسومات UML	
23	<i>Use case diagram</i>
25 : 24	<i>Sequence diagram</i>
26	<i>Class diagram</i>
28 : 27	<i>Activity diagram</i>

الباب الثالث : تصميم النظام (47 : 29)

29	شاشة الدخول
29	شاشة التسجيل
30	شاشة الرئيسية
شاشات الحجوزات	
31	شاشة اضافة حجز
31	شاشة قائمة الحجوزات
32	شاشة تعديل الحجز
32	شاشة حذف الحجز
شاشات الطلبات	
33	شاشة اضافة طلب
33	شاشة قائمة الطلبات
34	شاشة تعديل طلب
34	شاشة حذف طلب
الشاشات الخاصة بقائمة الطعام	
35	شاشة عرض اقسام الطعام
35	شاشة التحكم باقسام قائمة الطعام
36	شاشة اضافة قسم لقائمة الطعام

36	شاشة حذف قسم من قائمة الطعام
	الشاشات الخاصة بعناصر قائمة الطعام
37	شاشة عرض عناصر قائمة الطعام
37	شاشة التحكم بعناصر قائمة الطعام
38	شاشة اضافة عنصر لقائمة الطعام
38	شاشة حذف عنصر من قائمة الطعام
	الشاشات الخاصة باقسام الطاولات
39	شاشة التحكم باقسام الطاولات
39	شاشة اضافة قسم للطاولات
40	شاشة حذف قسم للطاولات
	الشاشات الخاصة بالطاولات
41	شاشة التحكم بالطاولات
41	شاشة اضافة طاولة
42	شاشة حذف طاولة
	الشاشات الخاصة بالموظفين
43	شاشة قائمة الموظفين
43	شاشة اضافة موظف
44	شاشة تعديل موظف

44	شاشة حذف موظف
	الشاشات الخاصة بالتقارير
45	شاشة تقارير الموظفين
45	شاشة تقارير الحجوزات
46	شاشة تقارير الطلبات
46	شاشة تقارير الفواتير
	شاشات اخرى
47	شاشة الاعدادات
47	شاشة فريق العمل
(الباب الرابع : تنفيذ النظام (120 : 48)	
49 : 48	ملف <i>web.php</i>
54 : 50	ملفات <i>migrations</i>
59 : 55	ملفات <i>models</i>
60	ملفات <i>view , controllers</i> الخاصة <i>login</i>
61	ملفات <i>view , controllers</i> الخاصة <i>register</i>
64 : 62	ملفات <i>view , controllers</i> الخاصة <i>area section</i>
71 : 65	ملفات <i>view , controllers</i> الخاصة <i>reservation</i>

74 : 72	ملفات view , controllers tables الخاصة
78 : 75	ملفات view , controllers menu section الخاصة
85 : 79	ملفات view , controllers menu item الخاصة
94 : 85	ملفات view , controllers order الخاصة
100 : 95	ملفات view , controllers invoice الخاصة
108 : 101	ملفات view , controllers employee الخاصة
111 : 109	ملفات view , controllers employee report الخاصة
114 : 112	ملفات view , controllers reservation report الخاصة
117 : 115	ملفات view , controllers order report الخاصة
120 : 118	ملفات view , controllers invoice report الخاصة

الباب الاول: الهيكل العام للنظام

المبحث الاول : مقدمة عن المشروع بصفة عامة

تعتبر الانظمة الالكترونية من اهم الامور التى يحتاجها ويهتم بها اصحاب المشروعات اليوم لما لها من اثر ايجابى على استثماراتهم بما يعود عليهم بالنفع واختصار الوقت و الجهد وتحقيق الاهداف بكفاءة وفاعلية , وغير ذلك من الامور الهامة والمفيدة لهذه الانظمة , وهذه الفوائد تظهر خصيصا فى الانظمة التى تكون موجهة للاستعمال المباشر مع العملاء , مثل نظام ادارة المطعم.

* "نظام ادارة مطعم YAMIFOOD" : هو نظام يتم ادارته والتعامل معه بواسطة : المديرين- الكاشير-الويتر فقط , فلهذا النظام فوائد كثيرة لكل من اصحاب المشروع والعاملين فيه والعملاء , ومنها :

- 1- سهولة قيام الكاشير بعملية الحجز للعميل.
- 2- سهولة قيام الويتر بعملية الطلب للعميل.
- 3- سهولة قيام الكاشير بعملية استخراج الفاتورة للعميل.
- 4- متابعة الاعمال التى تتم على النظام من خلال التقارير التى يقدمها النظام للمديرين.

وبذلك نستطيع نرى ان من خلال هذا النظام يكون هناك دقة وتنظيم بالعمل , وبالتالي نصل الى ان هذا النظام يقدم فوائد تعود بالنفع الكبير على العملاء اولا وعلى اصحاب المشروع من مديرين وموظفين ثانيا.

المبحث الثانى : شرح عمل النظام

فى هذا المبحث سنتحدث عن العمليات التى يقوم بها النظام تفصيلاً وبشكل مرتب :

1- عملية الاذونات :

- بالنسبة للمدير : يتوافر للمدير جميع صلاحيات النظام ، حيث يستطيع المدير القيام :

- * عملية التسجيل لنفسه و لجميع مستخدمين النظام (الكاشير والويترا).

- * العمليات المتعلقة بالحجز والطلب للعملاء.

- * العمليات المتعلقة بالفواتير.

- * عملية اعداد التقارير.

- مع ملاحظة ان المدير يستطيع القيام بالإضافة - التعديل - الحذف - البحث لجميع العمليات التى سبق ذكرها طبقاً لما تحتويه كل عملية من اجراءات.

- وبالتالي يستطيع المدير معرفة تفاصيل كل عملية يقوم بها كل مستخدم من مستخدمين النظام (كاشير او ويترا).

- بالنسبة للكاشير : يكون للكاشير صلاحيات محدودة عكس المدير ، حيث يقوم :

- * العمليات المتعلقة بالحجز.

- * العمليات المتعلقة بالفاتورة للعميل بناءاً على طلبه.

- مع ملاحظة ان الكاشير يقوم بالإضافة-التعديل-الحذف-البحث فى اي عملية يقوم بها.

- بالنسبة للويترا : يكون للويترا صلاحيات محدودة عكس المدير ، حيث يقوم :

- * العمليات المتعلقة بالطلب للعملاء.

- مع ملاحظة ان الويترا يقوم بالإضافة-التعديل-الحذف-البحث فى اي عملية يقوم بها.

2- عملية التسجيل التى يقوم بها المدير للقائمين على النظام :

- يقوم المدير بتسجيل بيانات العاملين على النظام مع تحديد وظيفة كل عامل وذلك بهدف السماح للعاملين بالدخول والتعامل مع النظام ، وما هى العمليات التى يقومون بها ، مع ملاحظة ان لكل مستخدم للنظام صلاحية معينة طبقاً لوظيفته.

3- العمليات الخاصة بقائمة الطعام و الطاولات :

- يقوم المدير بالتحكم فى قائمة الطعام ، حيث يقوم باضافة اقسام الى القائمة واضافة عناصر جديدة داخل هذه الاقسام ويتم التعامل مع كل عنصر برقم مميز خاص به.

- كما يقوم بالتحكم فى الطاولات التى توجد بالمطعم , حيث يقوم باضافة اقسام للطاولات مثل :

* قسم للمدخنين.

* قسم لغير المدخنين.

* قسم للمناسبات الخاصة.

* قسم المكان المفتوح.

- و اضافة عدد للطاولات داخل كل قسم ، ويتم التعرف على كل طاولة برقم مميز.

4- عملية الحجز :

- يقوم الكاشير بعملية الحجز للعملاء حيث يقوم بادخال بيانات العملاء على النظام كما يحق للعميل اختيار قسم معين بالمطعم.

5- عملية الطلب :

- يقوم الويتر بتسجيل بيانات طلب العميل على النظام بعد ان يختار العميل من قائمة الطعام ما يشاء من مأكولات او مشروبات، الى اخره من اقسام القائمة المختلفة.

- يقوم الويتر بعملية الطلب للعميل دون الزامه القيام بعملية الحجز.

6- عملية الدفع :

- يقوم الكاشير بعمل **checkout** للعميل فيخرج له فاتورة الطلب بناء على رقم طاولته، ثم يقوم العميل بالدفع للكاشير بأى وسيلة دفع.

7- عملية التقارير:

يقوم النظام بتقديم تقارير للمديرين عن جميع العمليات التى تتم بالنظام بناءا على طلبه.

المبحث الثالث : المتطلبات الرئيسية للمشروع

نقصد بالمتطلبات الرئيسية هى البيانات الاساسية التى لابد ان تكون موجودة ومسجلة بالنظام لىستطيع النظام تقديم خدماته ، وهذه البيانات هى :

1- بيانات المديرين:

- يملك المدير جميع صلاحيات النظام كما سبق الذكر تفصيليا.

2- بيانات الكاشير والویتر :

- من خلال الكاشير او الویتر يقوم النظام بتقديم خدماته للعملاء ، يقوم الكاشير بعملية الحجز واستخراج الفواتير للعملاء ، ويقوم الویتر بعمل طلب للعميل ، وبالتالي لکى يتمكن الكاشير او الویتر القيام باعملاهم لابد من ان تكون بياناتهم مسجلة على النظام مسبقا من خلال المديرين.

- يستطيع المدير القيام بجميع العمليات التى يقوم بها الكاشير او الویتر، لأن للمدير جميع صلاحيات النظام ، كما سبق الذكر.

3- بيانات قائمة الطعام و الطاولات :

- لابد من وجود بيانات لقائمة الطعام من حيث اقسامها والعناصر التى توجد داخل هذه الاقسام.

- وكذلك الامر بالنسبة للطاولات ، يجب ان يكون مسجل بالنظام اقسام للطاولات مع عدد مناسب من الطاولات داخل كل قسم.

- وبالتالي من خلال تسجيل بيانات قائمة الطعام والطاولات يستطيع الكاشير والویتر القيام بعملية الحجز و الطلب واستخراج فاتورة للعميل.

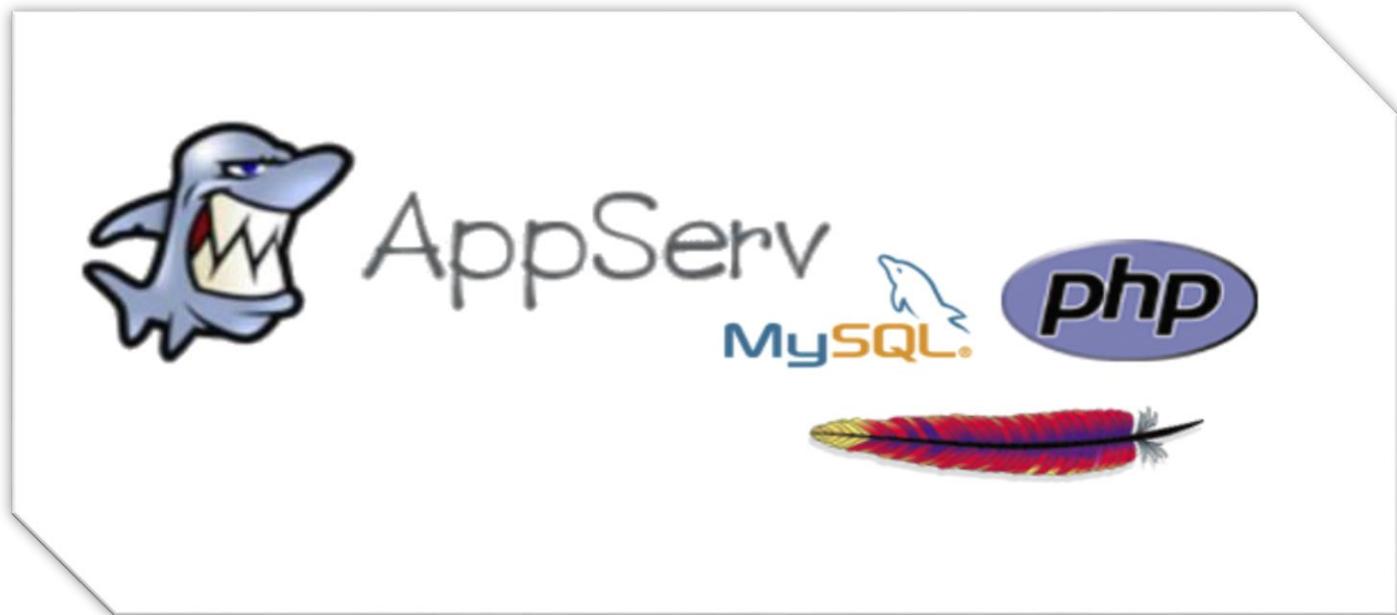
المبحث الرابع : اهداف المشروع

- 1 توفر الوقت والجهد للموظفين وللعملاء بالمطعم مما يساعد على اتمام الاعمال بالشكل المطلوب وبكفاءة وفاعلية.
- 2 تقديم مستوى عالى من الدقة و التنظيم فى العمل للقائمين على النظام.
- 3 العمل على ارضاء العملاء من خلال تقديم خدمات متميزة لهم.
- 4 تقديم تقارير للمديرين عن اي عملية بالنظام سواء كانت بيانات هذه العمليات متعلقة بالوقت الحالى او بالماضى.

المبحث الخامس : اللغات والبرامج المستخدمة

* البرامج المستخدمة :

: 1- برنامج (appserv)



- وقد تم تطوير هذا البرنامج بغرض توفير أداة تسهل وتساعد على تطوير التطبيقات الخاصة بـ MySQL و PHP والعديد من البرامج الأخرى.

- يوفر البرنامج لمستخدميه العديد من الخدمات، فبالإضافة إلى إمكانية تطوير التطبيقات المختلفة باستخدامه، فهو يسمح للمستخدم بإنشاء سيرفر كامل خاص به بشكل محلي على جهازه الخاص لاستخدامه بغرض فحص ومعاينة صفحات وتطبيقات النت المطورة من قبله قبل رفعها على سيرفر فعلي وإتاحتها للعموم وهو الاستخدام الأشهر للبرنامج.

- يأتي برنامج AppServ بالكثير من الخدمات والبرامج المضمنة وهي (Apache , PHP , MySQL , phpMyAdmin , TSL – SSL – HTTPS protocols).

:(composer) 2 - برنامج



- عبارة عن أداة لإدارة المشاريع البرمجية في PHP تسمح لك بتحديد المكتبات البرمجية التي يعتمد عليها مشروعك و Composer يتولى إدارة المشروع من تثبيت و تحديث المكتبات لك مثل Laravel .

- قوم بتنصيب المكتبات او الحزم داخل مجلد بمسماي vendor داخل مجلد مشروعك. فلن يقوم Composer بتنصيب اي مكتبه بمسار عام بجهازك ابدا بشكل افتراضي. ولهذا سمي dependency manager .

3- برنامج (visual studio code)



- هو محرر شيفرات مجاني و مفتوح المصدر "MIT License" مطور من قبل Microsoft موجه للعديد من الانظمة . "Windows, Linux and macOS" يدعم هذا المحرر العديد من اللغات ويحتوي على عدد كبير من الاضافات التي تسهل عليك العمل من كتابة و هيئة المحرر و تحليل الشفرة . "Code" في عام 2018 حصل محرر Visual Studio Code على المركز الاول في استبيان stackoverflow كأكثر بيئة تطوير مستخدمة بين المطورين.

- يدعم JavaScript, TypeScript, CSS, HTML ولغات اخرى عديدة ويمكنك دعم لغات اكثرا عن طريق VS Code وكما يوفر توثيق للتعامل مع بعض اللغات يمكن ان تستكشفها في التوثيق Marketplace الرسمي للمحرر.

: (erdplus) 4 - موقع

من خلال هذا الموقع تم انشاء رسومات Data Base وهم: .ERD, RS

**: (visual paradigm) 5 - برنامج**

- من خلال هذا البرنامج يتم صنع الكثير من الرسومات ذات الانواع المختلفة ، ولكن ما يهمنا هنا ، ان من خلال هذا البرنامج تم عمل رسومات ومخططات

UML diagrams مثل: Class, activity ,Use case, sequence



* اللغات المستخدمة :

1 - اطار عمل (Laravel 7)



لارافل Laravel هو اطار عمل خاص بلغة PHP سيجعلك تقع حرفياً في حب هذه اللغة حتى وان كنت مصمم في الاساس ، يقدم لارافل بيئة عمل متكاملة لكل ما تحتاجه او قد تحتاجه في اي فترة مستقبلية من حياتك المهنية ، لارافل Laravel مبني على طريقة MVC الشهيرة التي تفصل ال View عن Model Controller مما يوفر لك بيئة عمل سلسة وسهلة ، وربما لحد الان لم تستطع استيعاب ماهية لارافل Laravel تماماً ، لا داع للقلق فالسطور القادمة كفيلة بمساعدتك للدخول الى بداية مشوارك في تعلم لارفيل ، ولكن يتم تسطيبها تحتاج الى برنامج : appserv . composer

2 - برنامج (MySQL)

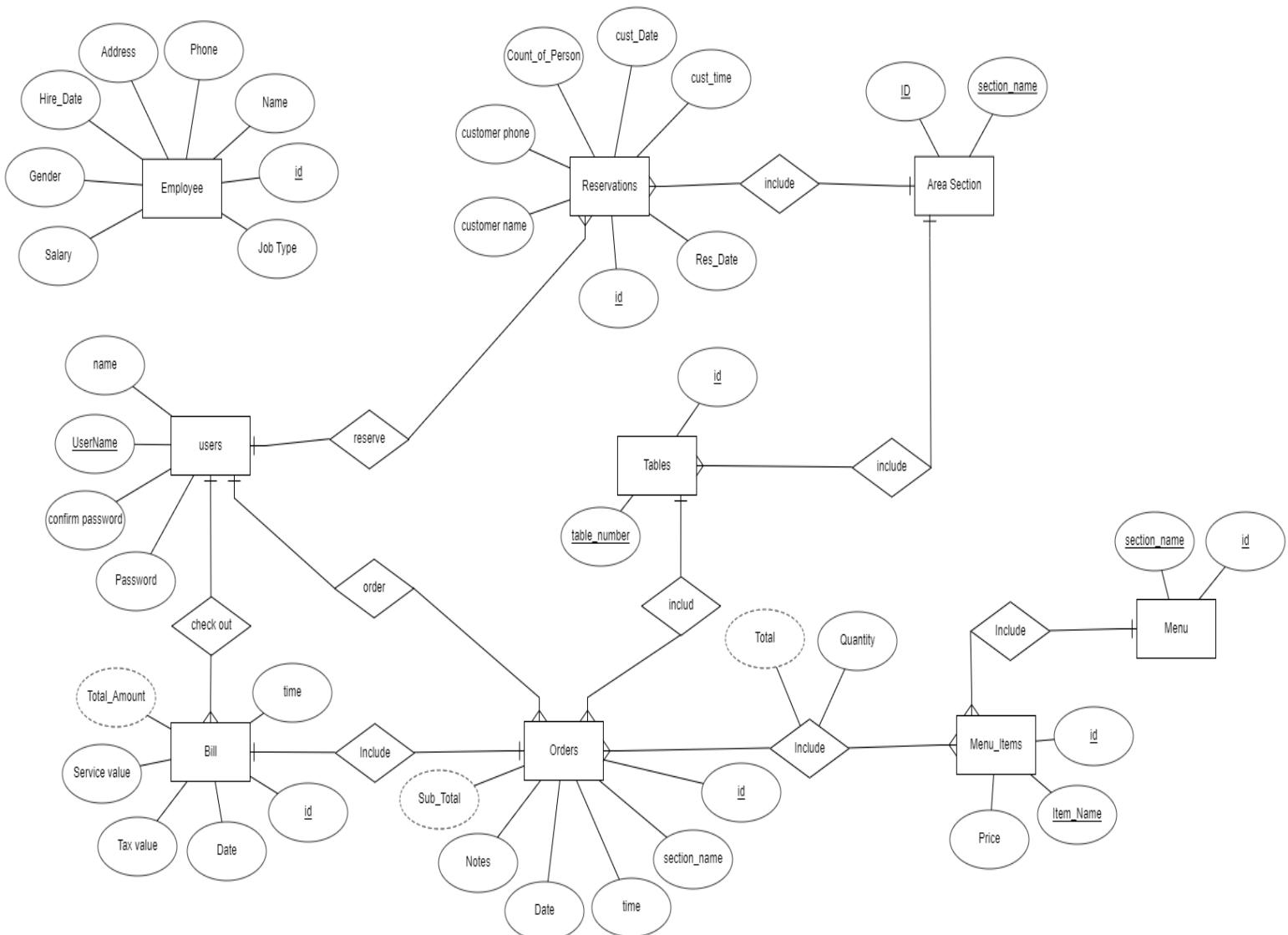


هو برنامج لإدارة قواعد البيانات Database Management System, DBMS يُساعد مستخدميه في تخزين ، تنظيم والعثور على البيانات. ينتشر استخدام MySQL في موقع الويب نظرًا لميزاته والمرونة التي يُوفرها. نهدف في هذا الدرس تقديم أساسيات استخدام MySQL بطريقة ميسّرة وسهلة.

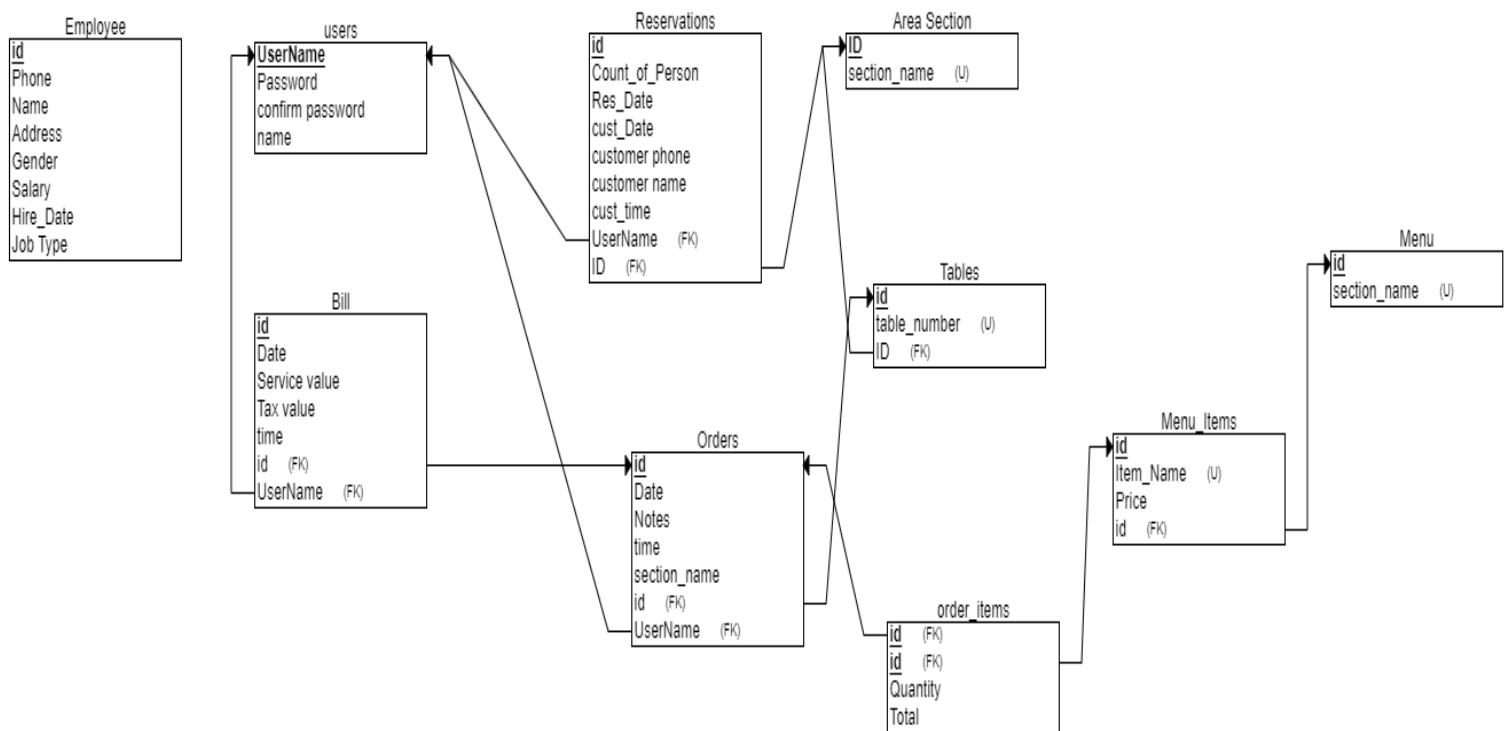
الباب الثانى : تحليل النظام

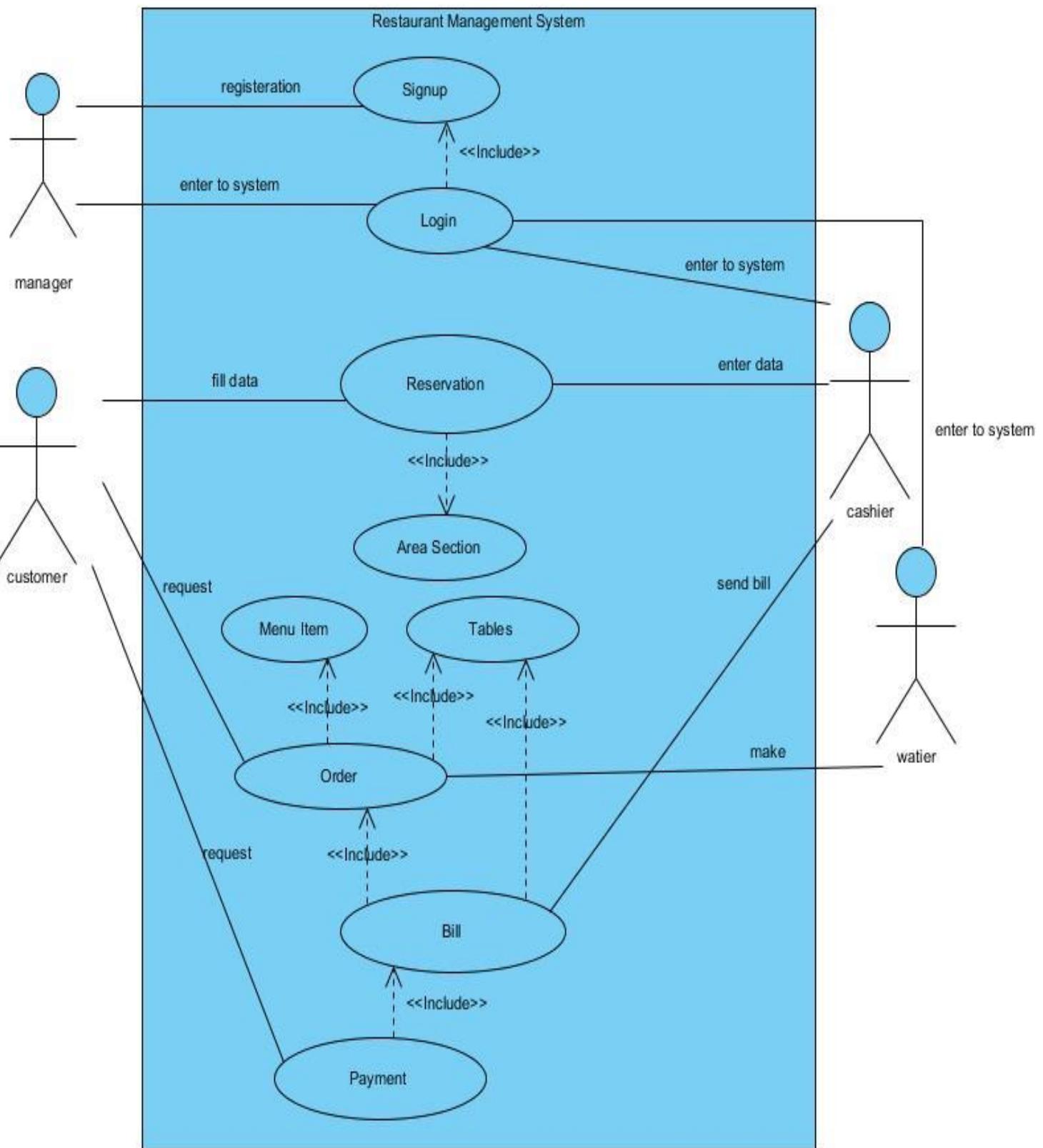
المبحث الاول : رسومات قاعدة بيانات

1-رسمة (ERD):

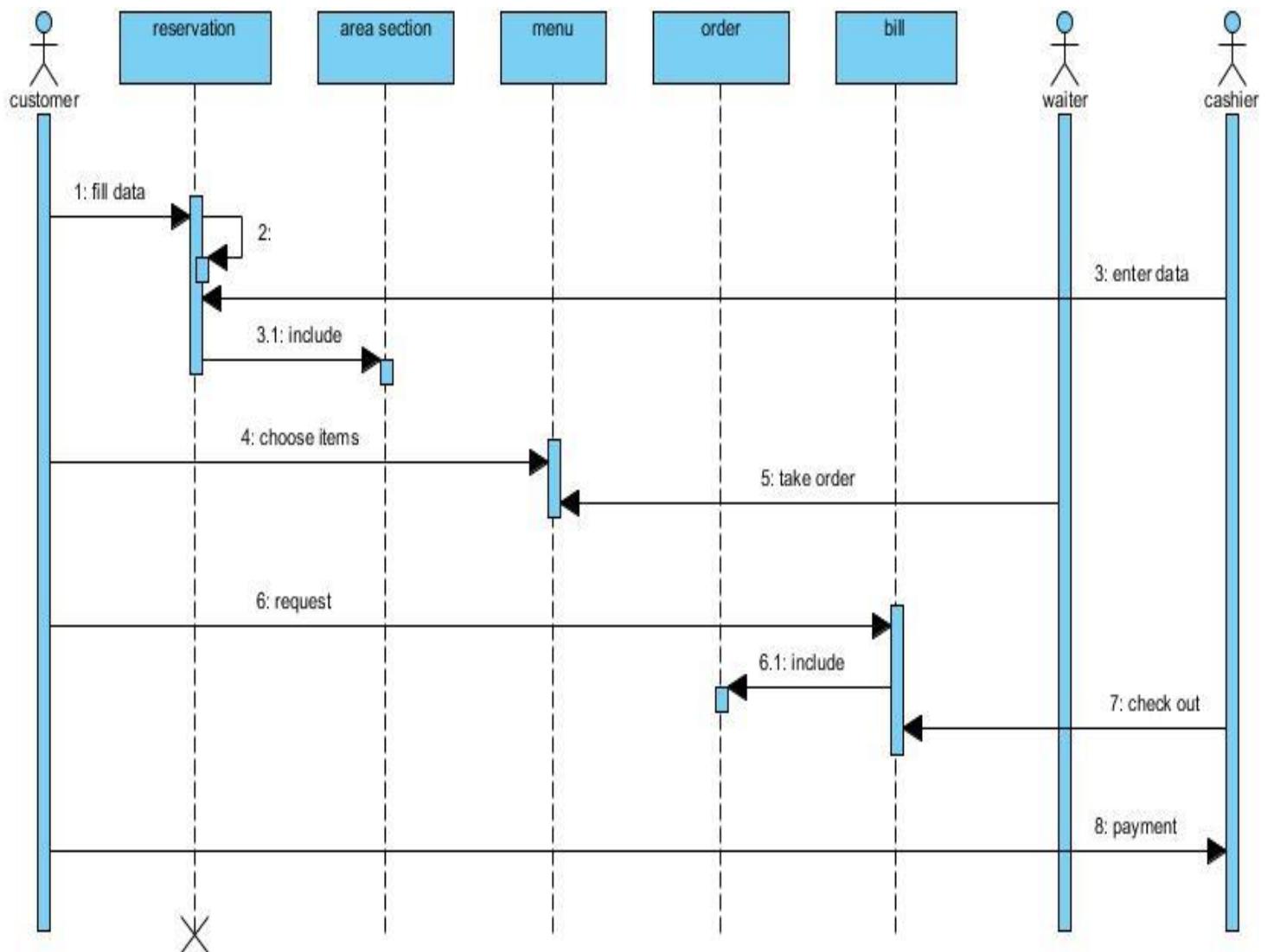


- رسمة : (RS) -2

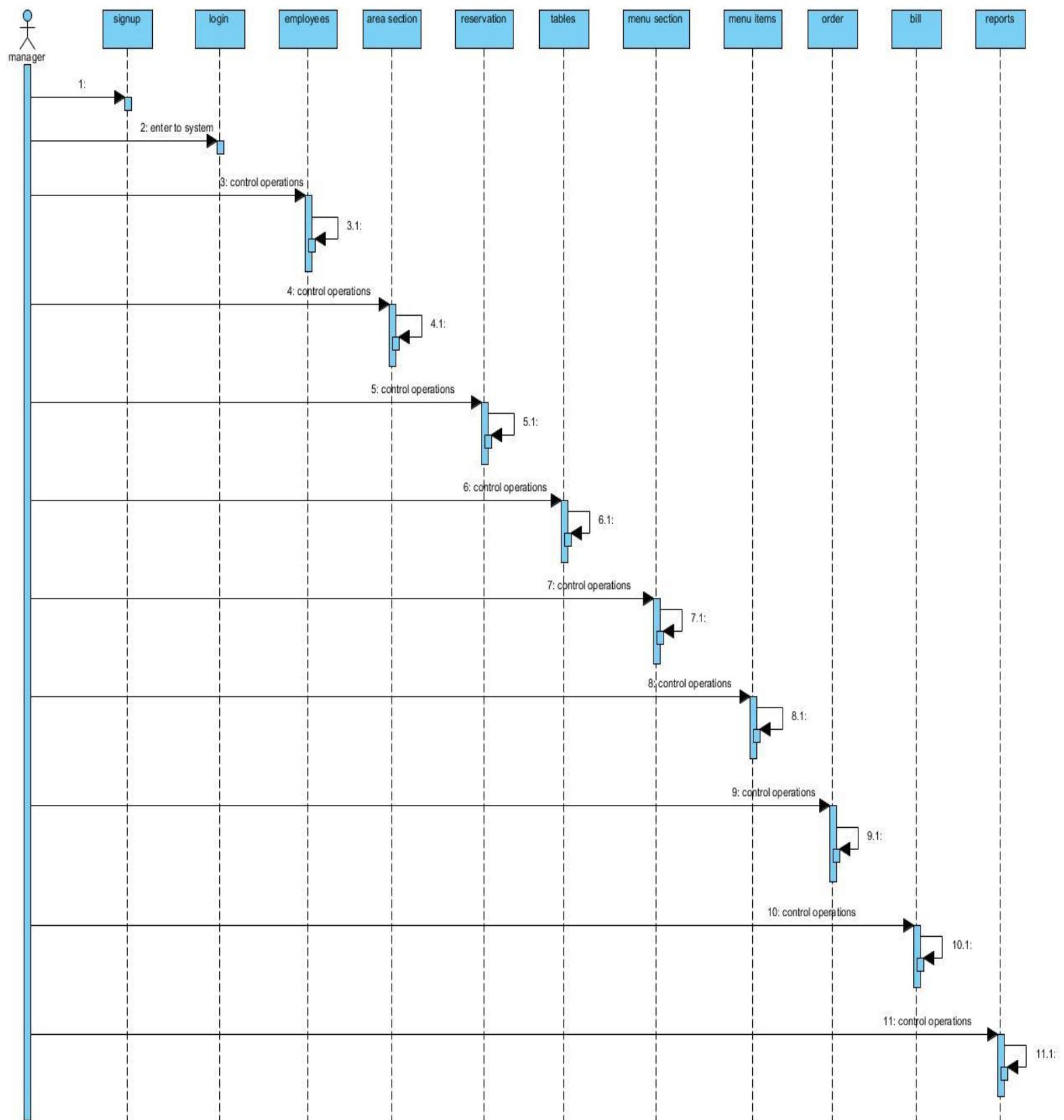


المبحث الثانى : رسومات UML**:(use case diagram) -1**

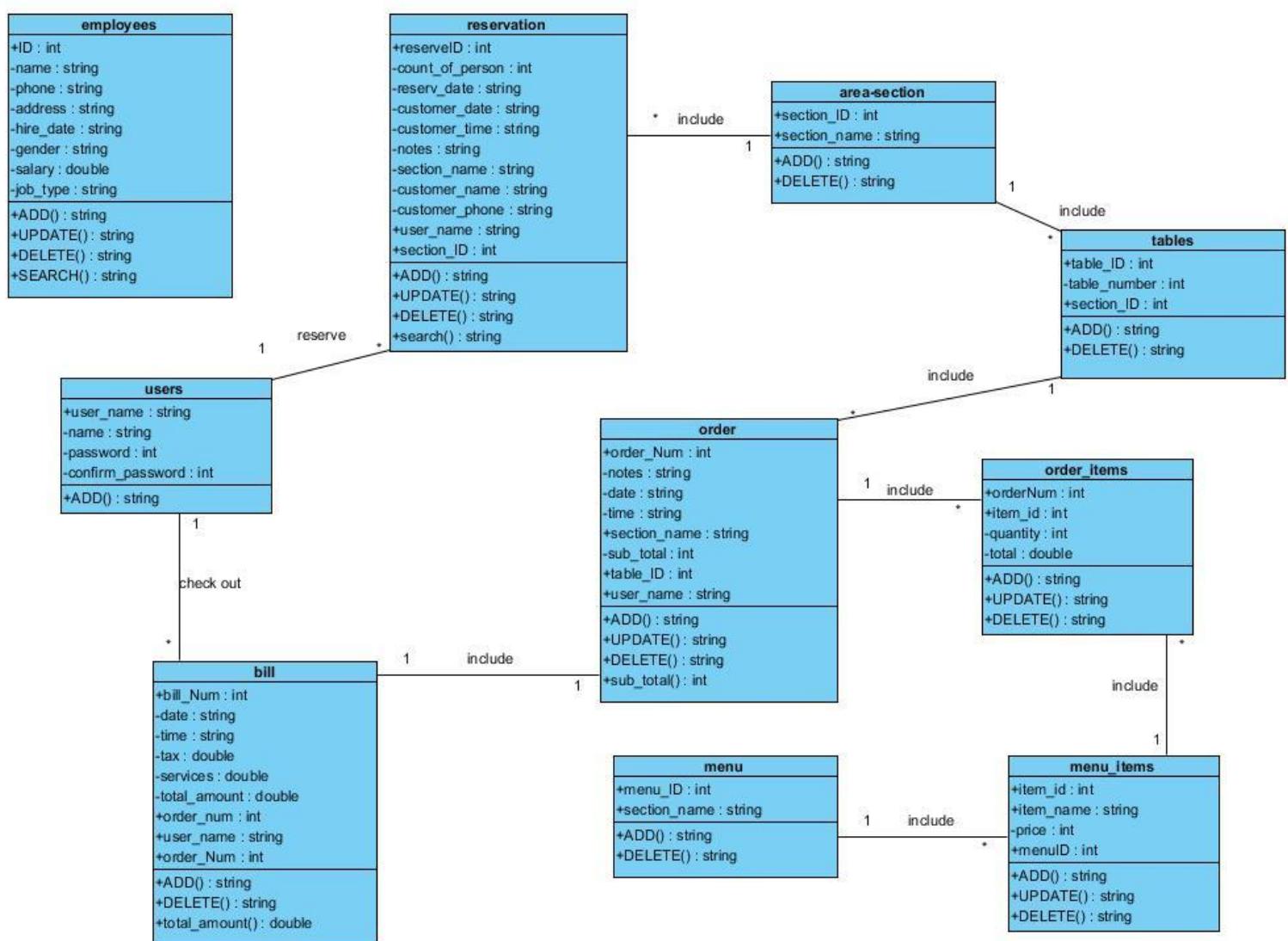
:(Sequence diagram) -2



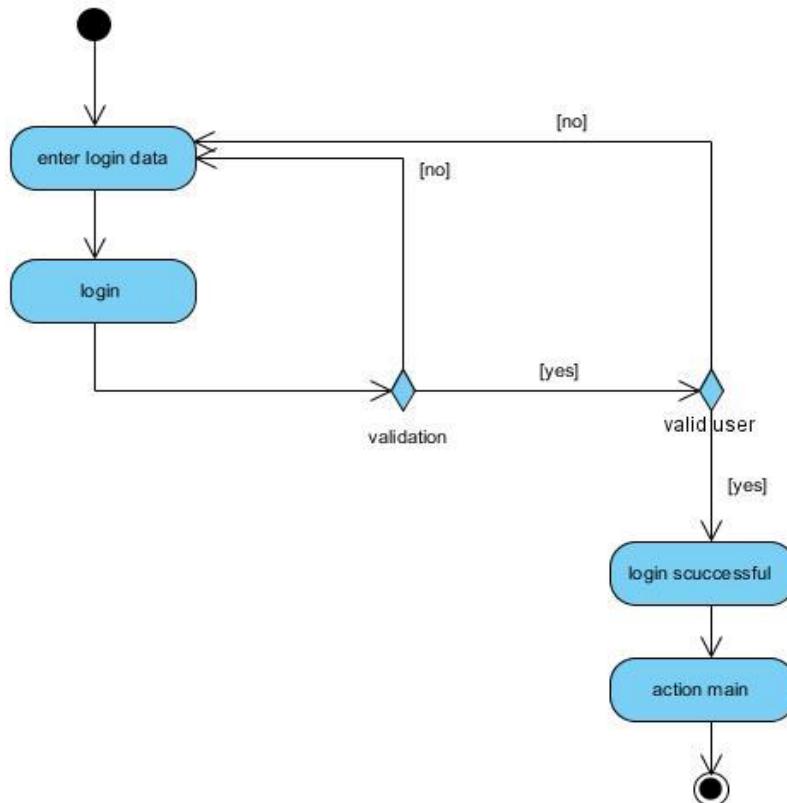
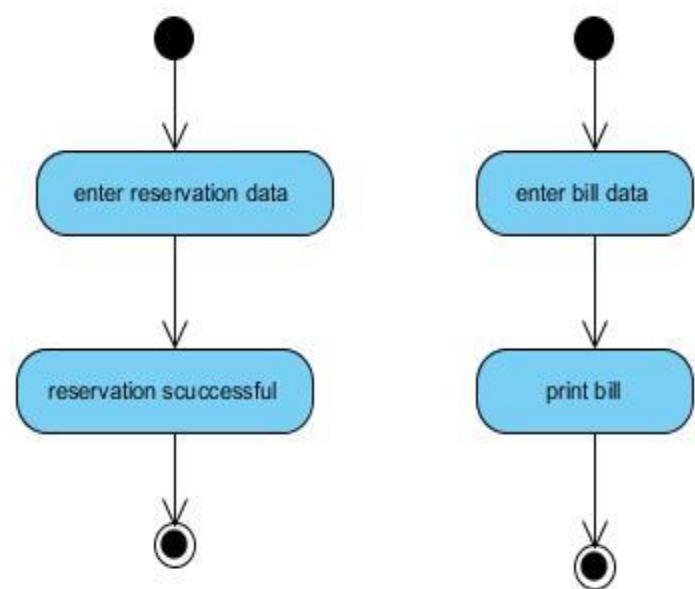
3-رسمة (Sequence diagram) للمدير :

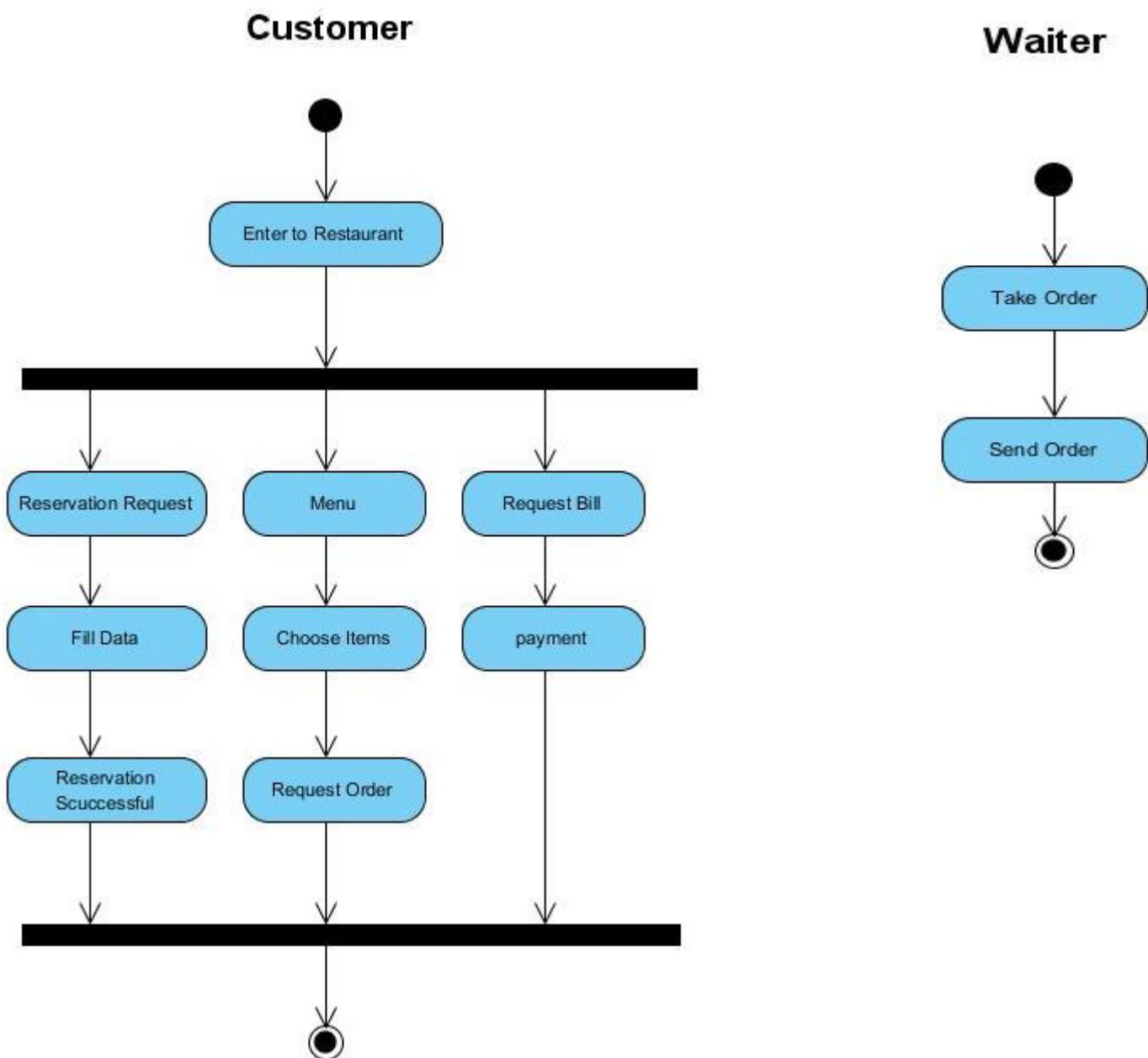


: (class diagram) -4



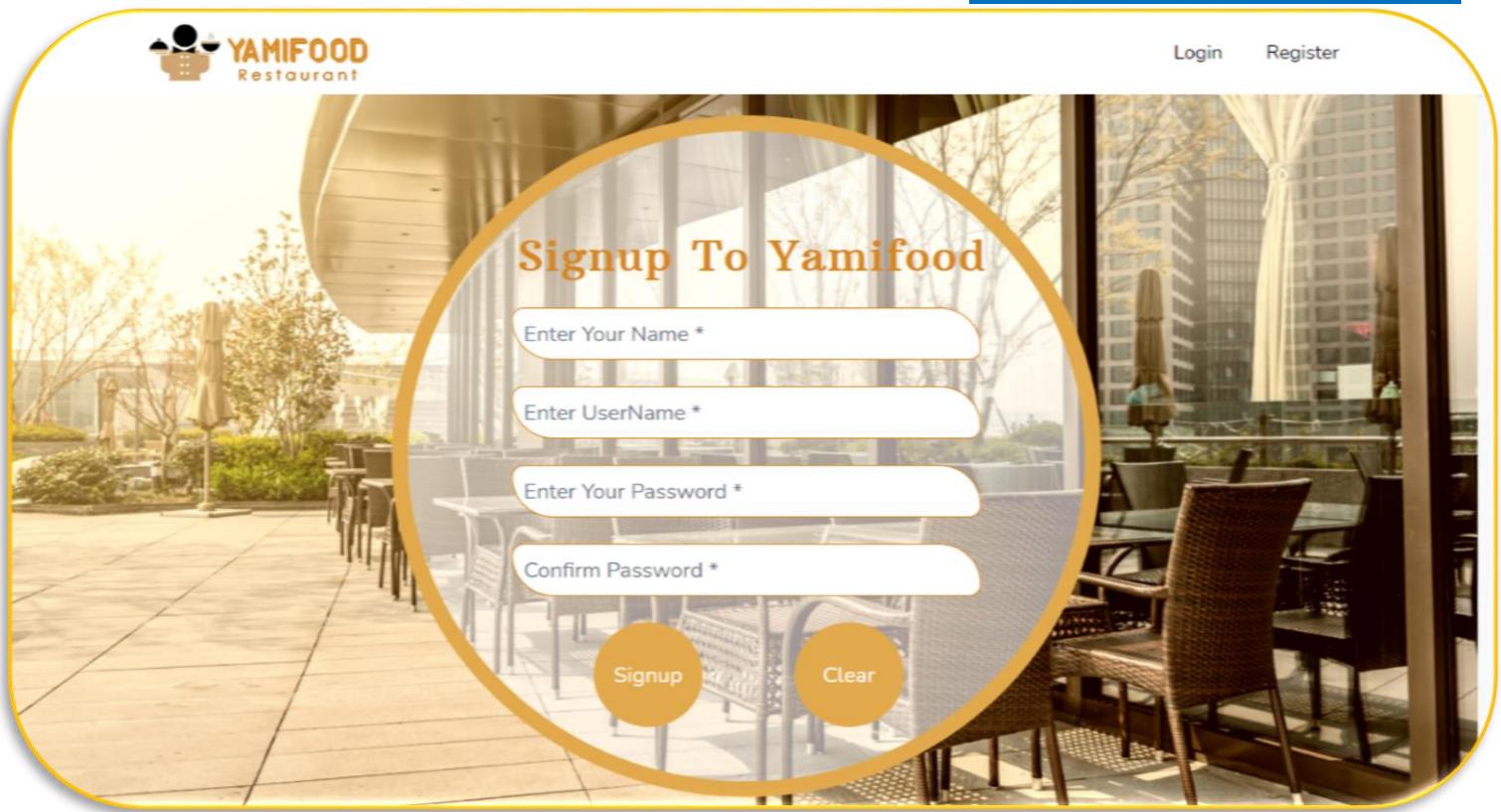
:(Activity diagrams) 5-رسومات

Login**Cahier**

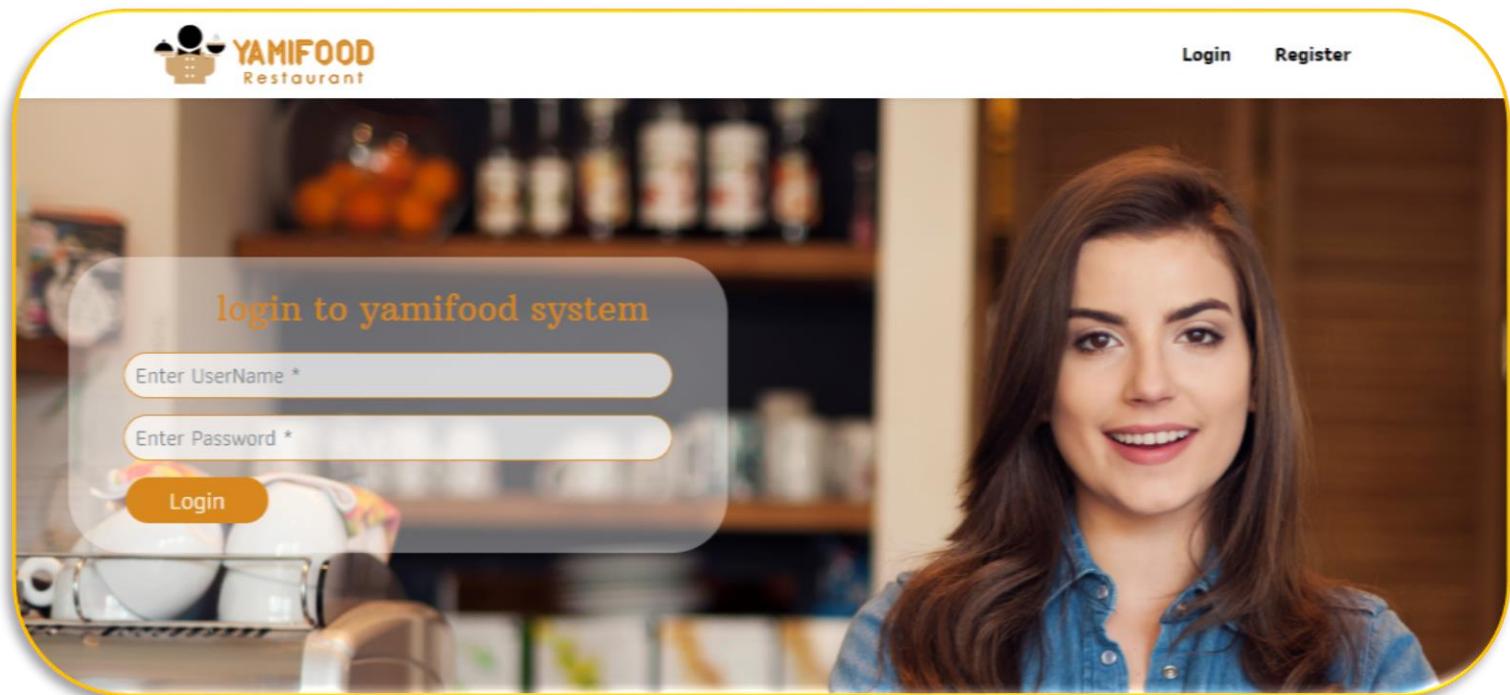


الباب الثالث : تصميم النظام

* شاشة التسجيل : Register



* شاشة الدخول : Login



*** الشاشة الرئيسية : Home page ***

YAMIFOOD Restaurant

- Home Reservation Orders Food Menu Settings Ar ▾

Welcome To Yamifood Restaurant

Yamifood Restaurant System That Helps Its Employees Work Efficiently, Accurately And Easily In The Shortest Time.



Welcome To Yamifood Restaurant

Little Story

Loreum Ipsum Dolor Sit Amet, Consectetur Adipiscing Elit. Pellentesque Auctor Suscipit Feugiat. Ut At Pellentesque Ante, Sed Convallis Arcu. Nullam Facilisis, Eros In Eleifend Luctus, Odio Ante Sodales Augue, Eget Lacinia Lectus Erat Et Sem.

Sed Semper Orci Sit Amet Porta Placerat. Etiam Quis Finibus Eros. Sed Aliquam Metus Lorem, A Pellentesque Tellus Pretium A. Nulla Placerat Elit In Justo Vestibulum, Et Maximus Sem Pulvinar.

" If You're Not The One Cooking, Stay Out Of The Way And Compliment The Chef. "

Michael Strahan

Presentation For Main Meals

Pumpkin soup

PRICE 10 \$

Yamifood Restaurant Owners

All Rights Reserved. © 2021 Yamifood Restaurant Designed And Developed By : ARM Team

الشاشات الخاصة بالحجوزات

* شاشة اضافة حجز Reservation

Customer Name *

Customer Phone *

Count Of Person *

06/25/2021

select section

select table

Mm/Dd/YYYY

--:-- --

Add

Clear

* شاشة قائمة الحجوزات Reservations Menu

ID	Customer Name	Customer Phone	Count Of Person	Area Place	Table Number	Reserved Date	Customer Date	Customer Time	User Name	Update	Delete
1	Mohamed Marouf	01234567891	5	Smoking	11	2021-06-24	2021-06-24	21:53:00	Mohamed		

*** شاشة تعديل الحجز : Reservation update**

mohamed marouf

01234567891

5

2021-06-24

smoking

11

06/24/2021

09:53 PM

Update

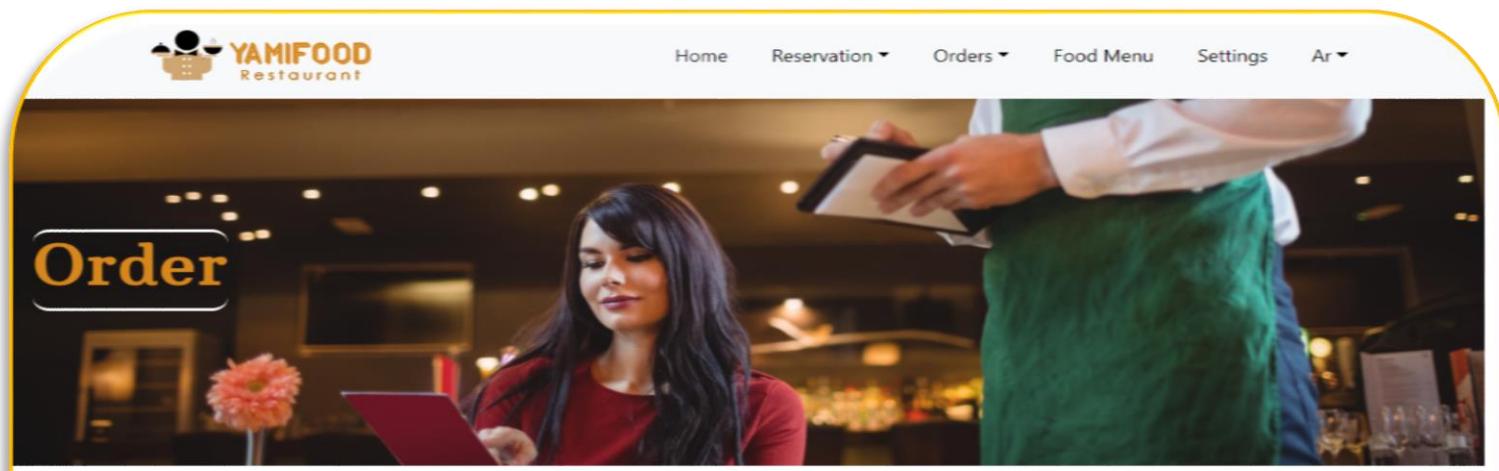
Clear

*** شاشة حذف الحجز : Reservation delete**

ID	Customer Name	Customer Phone	Court Per	Non Smoking	22	2021-06-26	2021-06-25	Customer Date	User Name	Update	Delete
1	Ahmed Omar	01234567891	6	Non Smoking	22	2021-06-26	2021-06-25		Ar	<input checked="" type="checkbox"/>	<input type="checkbox"/>

الشاشات الخاصة بالطلب

* شاشة اضافة طلب : Order



The screenshot shows the 'Order' screen of the Yamifood restaurant management system. At the top, there's a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and Language (Ar). On the left, a large button labeled 'Order' is visible. The main area contains several dropdown menus for selecting section, table, menu, and menu item. To the right, there are fields for date (25-Jun-2021), time (04:03:19 PM), and notes (any notes). Below these are buttons for sub-total, Save, Clear, Item Name, Price, Quantity, Total, and Ignore. A footer bar at the bottom includes links for Home, Reservation, Orders, Food Menu, Settings, and Language (Ar).

* شاشة قائمة الطلبات : Orders Menu



The screenshot shows the 'Orders Menu' screen. At the top, there's a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and Language (Ar). Below the navigation is a search bar with placeholder text 'search for orders' and a 'Search' button. The main area features a heading 'Order Controls' and a table displaying a list of orders. The table has columns for ID, Section Name, Table Number, Date, Time, Notes, User Name, Update, and Delete. The first row of data in the table is:

ID	Section Name	Table Number	Date	Time	Notes	User Name	Update	Delete
7	Non Smoking	22	2021-06-25	16:04:34		Ar	<input checked="" type="checkbox"/>	<input type="checkbox"/>

* شاشة تعديل الطلب : Order update

* شاشة حذف الطلب : Order delete

الشاشات الخاصة بقائمة الطعام*** شاشة عرض اقسام قائمة الطعام : Menu Section**

The screenshot shows a web-based application interface for a restaurant named 'YAMIFOOD Restaurant'. At the top, there is a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and Language selection (Ar). Below the navigation, a large orange header box contains the text 'Menu Sections'. Underneath this, there are three circular icons, each featuring a chef in a kitchen setting. The first icon is labeled 'Hot Drinks', the second 'Cold Drinks', and the third 'Desserts'.

*** شاشة التحكم باقسام قائمة الطعام : Menu Section Control**

The screenshot shows a 'Menu Controls' screen. At the top, there is a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and Language selection (Ar). Below the navigation, a large orange header box contains the text 'Menu Controls'. A green button labeled '+ Add New Section' is visible. Below the button is a table listing three menu sections: 'Hot Drinks', 'Cold Drinks', and 'Desserts', each associated with an ID (1, 2, 3) and a red delete icon.

ID	Section Name	Delete
1	Hot Drinks	
2	Cold Drinks	
3	Desserts	

* شاشة اضافة قسم لقائمة الطعام : Menu New Section

ID		Delete
1	Hot Drinks	
2	Cold Drinks	
3	Desserts	

* شاشة حذف قسم من قائمة الطعام : Menu Section Delete

ID		Delete
1	Hot Drinks	
2	Cold Drinks	
3	Desserts	

الشاشات الخاصة بعناصر قائمة الطعام*** شاشة عرض عناصر قائمة الطعام : Menu Items**

The screenshot shows a web-based application interface for a restaurant named 'YAMIFOOD Restaurant'. At the top, there is a navigation bar with links for 'Home', 'Reservation ▾', 'Orders ▾', 'Food Menu', 'Settings', and 'Ar ▾'. Below the navigation bar, the word 'ITEMS' is displayed in a large, bold, red font inside a rounded rectangular box. Below this box are four red circular buttons, each containing a white text label: 'Tea', 'Cofee', 'Hot Chocolate', and 'Milk'.

*** شاشة التحكم بعناصر قائمة الطعام : Menu Items Control**

The screenshot shows a web-based application interface for a restaurant named 'YAMIFOOD Restaurant'. At the top, there is a navigation bar with links for 'Home', 'Reservation ▾', 'Orders ▾', 'Food Menu', 'Settings', and 'Ar ▾'. Below the navigation bar, the title 'Menu Items Controls' is displayed in a large, bold, orange font inside a rounded rectangular box. At the bottom left, there is a green button labeled 'Add New Item' with a plus sign icon. Below the title, there is a table with the following data:

Item Name	Price	Section Name	Update	Delete
Tea	5	Hot Drinks	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Coffee	10	Hot Drinks	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Hot Chocolate	25	Hot Drinks	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Milk	5	Hot Drinks	<input checked="" type="checkbox"/>	<input type="checkbox"/>

* شاشة اضافة عنصر لقائمة الطعام : Menu New Item

Add Item

Item Name
Price
Section Name
Select Section

SAVE CLOSE

Item Name	Update	Delete
Tea	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Coffee	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Hot Chocolate	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Milk	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>

* شاشة حذف عنصر بقائمة الطعام : Menu Item Delete

Delete Item

Do You Sure Delete Item

Item Name
tea

DELETE CLOSE

Item Name	Update	Delete
Tea	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Coffee	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Hot Chocolate	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>
Milk	<input checked="" type="checkbox"/>	<input type="button" value="Delete"/>

الشاشات الخاصة باقسام الطاولات*** شاشة التحكم باقسام الطاولات : Area Section Control**

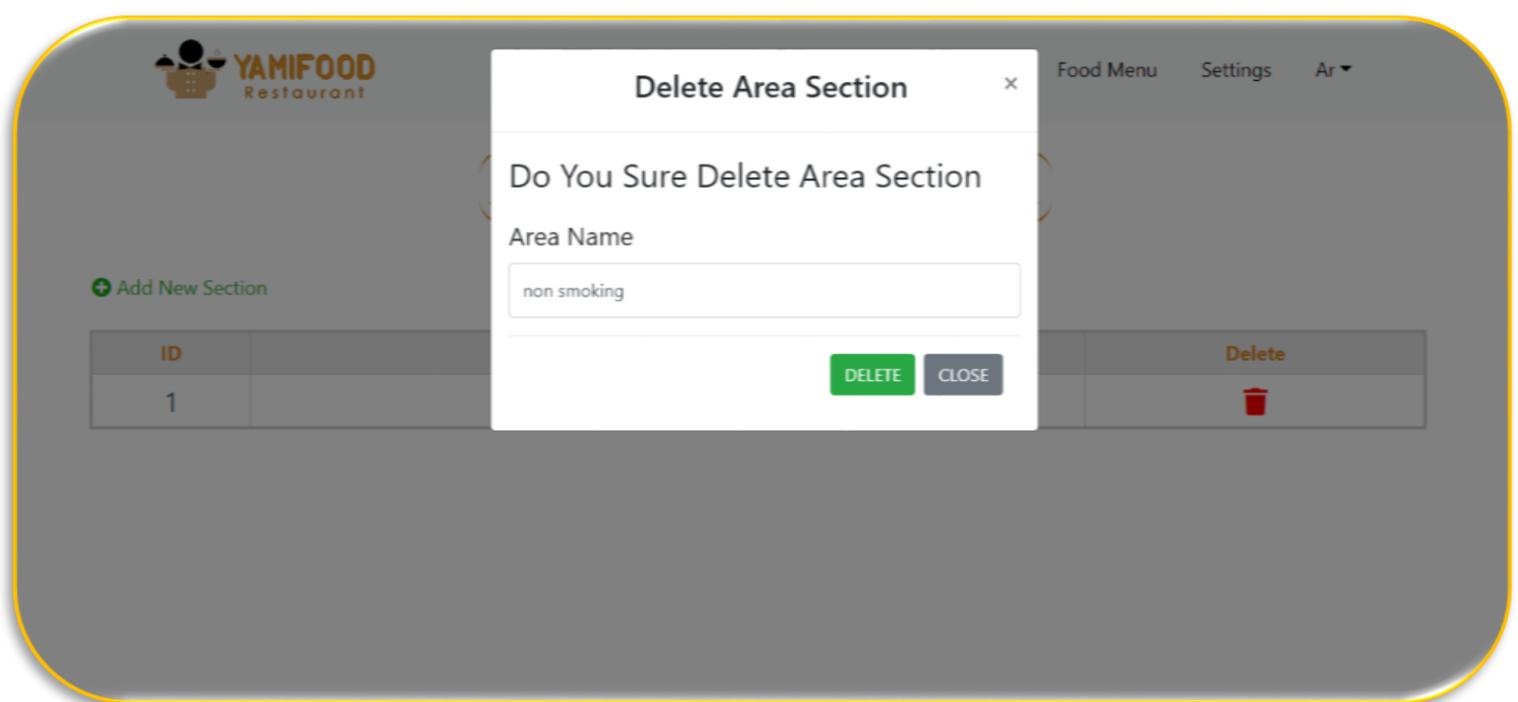
The screenshot shows a web-based application interface for managing table sections. At the top, there is a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and Language (Arabic). The main title 'Area Section Control' is displayed prominently. Below the title is a button labeled '+ Add New Section'. A table lists four existing sections: Non Smoking (ID 1), Smoking (ID 2), Special (ID 3), and Open (ID 4). Each row in the table includes a 'Delete' button represented by a red trash icon.

ID	Area Section	Delete
1	Non Smoking	
2	Smoking	
3	Special	
4	Open	

*** شاشة اضافة قسم للطاولات : Area New Section**

The screenshot shows a modal dialog box titled 'Add Area Section'. It contains a form with a label 'Area Name' and a text input field. At the bottom of the dialog are two buttons: 'SAVE' (green) and 'CLOSE' (grey). In the background, the main application interface is visible, showing a table with one section entry: ID 1, Non Smoking. There is also a '+ Add New Section' button.

* شاشة حذف قسم للطاولات : Area Delete Section



الشاشات الخاصة بالطاولات*** شاشة التحكم بالطاولات : Tables Control**

The screenshot shows a table titled "Tables Controls" with the following data:

Table Number	Area Section	Delete
22	Non Smoking	
15	Smoking	
55	Special	
10	Open	

[+ Add New Table](#)

*** شاشة اضافة طاولة : New Table**

The screenshot shows a modal dialog box titled "Add Table Number" with the following fields:

- Table Number:
- Section Name:
- Buttons: **SAVE** and **CLOSE**

The background shows a list of existing tables:

Table Number	Area Section	Delete
22	Non Smoking	
15	Smoking	
55	Special	
10	Open	

[+ Add New Table](#)

*** شاشة حذف طاولة : Delete Table**

The screenshot shows the Yamifood Restaurant management system interface. At the top, there is a logo for 'YAMIFOOD Restaurant' and navigation links for 'Food Menu', 'Settings', and language selection ('Ar'). A central modal dialog box titled 'Delete Table' asks 'Do You Sure Delete Table Number'. Below the dialog, a table lists tables with numbers 22, 15, 55, and 10. To the right of each table number are columns for 'Smoking', 'Special', and 'Open' status, along with a 'Delete' button represented by a trash can icon.

Table Number	Smoking	Special	Open	Delete
22				
15				
55				
10				

الشاشات الخاصة بالموظفين*** شاشة قائمة الموظفين : Employees Menu**

The screenshot shows a web application interface for managing employees. At the top, there is a navigation bar with links for Home, Reservations, Orders, Food Menu, Settings, and a user account for Mohamed. Below the navigation bar, a large orange header box contains the text "Employees Controls". Underneath this, there is a search bar with the placeholder "search for employees" and a "Search" button. To the right of the search bar is a green "Add New Employee" button with a plus sign icon. Below the search area is a table with the following columns: ID, Employee Name, Phone, Address, Hire-Date, Salary, Job-Type, Gender, Update, and Delete. A single row of data is shown:

ID	Employee Name	Phone	Address	Hire-Date	Salary	Job-Type	Gender	Update	Delete
1	Mohamed Ahmed	01234567899	Giza Haram	2021-06-25	3000	Manager	Male		

*** شاشة اضافة موظف : New Employee**

The screenshot shows a web application interface for adding a new employee. At the top, there is a navigation bar with links for Home, Reservation, Orders, Food Menu, Settings, and language options (Arabic). Below the navigation bar, a large orange header box contains the text "Employees". On the left side of the page is a photograph of three restaurant staff members (two women and one man) standing in a dining area. To the right of the photograph are several input fields for entering employee information. The fields include:

- name *
- phone *
- address *
- 2021-06-25
- salary *
- select job type
- select gender

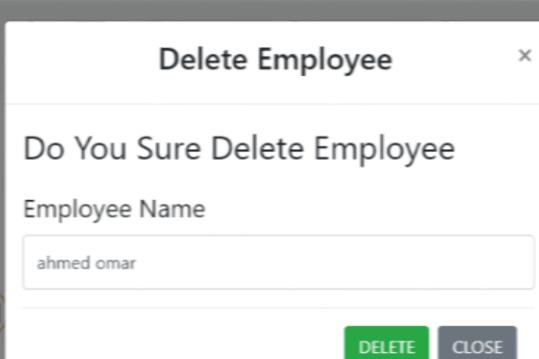
At the bottom of the form are two buttons: "Save" and "Clear". There is also a small orange arrow icon pointing upwards in the bottom right corner.

*** شاشة تعديل موظف Employee Update**


The screenshot shows the 'Employees Update' page. On the left, there is a photograph of three staff members (two women and one man) in a restaurant. On the right, there are several input fields for updating employee information:

- Employee Name: ahmed omar
- Phone: 01203020512
- Address: giza faysel
- Date of Birth: 24-Jun-2021
- Salary: 2000
- Job Type: cashier
- Gender: male

At the bottom are 'Update' and 'Clear' buttons, and a small upward arrow icon.

*** شاشة حذف موظف Employee Delete**


The screenshot shows a 'Delete Employee' dialog box with the following content:

Delete Employee

Do You Sure Delete Employee

Employee Name: ahmed omar

Buttons: DELETE (green), CLOSE (grey)

Background: A list of employees is visible, showing one entry:

ID	Employee Name	Phone	Join Date	Job-Type	Gender	Update	Delete		
1	Ahmed Omar	01203020512	Giza Faysel	2021-06-24	2000	Cashier	Male		

الشاشات الخاصة بالتقارير* شاشة تقارير الموظفين : Employees Reports

The screenshot shows the 'Employee Report' section of the Yamifood restaurant management system. At the top, there is a navigation bar with links for Home, Reservations, Orders, Food Menu, Settings, and Mohamed. Below the navigation is a search bar with fields for 'search for employees', 'Select All', date ranges, and a 'Search' button. To the right of the search bar are download and export icons. A table below the search bar displays columns for ID, Employee Name, Phone, Address, Hire-Date, Salary, Job-Type, and Gender. A message 'No Matching Records Found' is centered in the table area.

* شاشة تقارير الحجوزات : Reservations Reports

The screenshot shows the 'Reservations Report' section of the Yamifood restaurant management system. It features a similar navigation bar at the top. Below the navigation is a search bar with fields for 'User Name', 'Select All', 'Customer Name', 'Customer Phoe', date ranges, and a 'Search' button. To the right are download and export icons. A table below the search bar displays columns for ID, Customer Name, Customer Phone, Count Of Person, Area Place, Table Number, Customer Date Time, Reserved Date Time, Notes, and User Name. The table contains several rows of sample data.

*** شاشة تقارير الطلبات : Orders Reports**
[Home](#) [Reservations](#) [Orders](#) [Food Menu](#) [Settings](#) [Mohamed](#)
Orders Report

Table Number

Select All

mm/dd/yyyy



mm/dd/yyyy



Search



ID	Section Name	Table Number	Date	Time	Notes	Sub Total	User Name
Asasas	Asasas	Asasas	Asasas	Asasas	Asasas	Asasas	Asasas

*** شاشة تقارير الفواتير : Bills Reports**
[Home](#) [Reservations](#) [Orders](#) [Food Menu](#) [Settings](#) [Mohamed](#)
Invoices Report

Table Number

Select All

mm/dd/yyyy



mm/dd/yyyy



Search



ID	Order Number	Table Number	Date And Time	Service Value	Tax Value	Total Amount	User Name
Asasas	Asasas	Asasas	Asasas	Asasas	Asasas	Asasas	Asasas

*** شاشة الاعدادات :Settings**
*** شاشة فريق العمل :Team Work**

Development Team

Abdul Rahman Ali Mohamed Marof Mahmoud Yacoup Salma Adel

الباب الرابع : تنفيذ النظام

- ملف web.php

```

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use Illuminate\Support\Facades\Auth;
5 /*
6 |-----|
7 | Web Routes
8 |-----|
9 |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 // يقوم بجعل صفحة تسجيل الدخول هي الصفحة الاولى التي تظهر في النظام مع القيام ايضاً بعرضها
17 Route::get('/', function () {
18     return view('auth.login');
19 });
20
21 // بمجرد الضغط على الرابط الخاص بصفحة فريق العمل يقوم مباشرة بفتح هذه الصفحة
22 Route::get('/dev-team', function () {
23     return view('dev-team');
24 });
25
26 // بمجرد الضغط على الرابط الخاص بصفحة الاعدادات يقوم مباشرة بفتح هذه الصفحة
27 Route::get('/settings', function () {
28     return view('settings');
29 });

```

```

1 -----start employees pages-----
2
3 Route::get('/search_emp', 'EmployeeController@search_emp');
4
5
6 // وظيفته تشغيله على السيرفر
7 // Route resource سواء اضافة او تعديل او مسح او بحث
8 Route::resource('/employees', 'EmployeeController');
9
10 -----end employees pages-----
11
12 -----start tables pages-----
13
14 // Route resource سواء اضافة او تعديل او مسح او بحث
15 Route::resource('/area-section', 'TableController');
16
17 // Route resource سواء اضافة او تعديل او مسح او بحث
18 Route::resource('/tables', 'tablecount');
19
20 -----end tables pages-----
21
22 -----start reservation pages-----
23
24 Route::get('/search_res', 'ReservationController@search_res');
25
26 // Route resource سواء اضافة او تعديل او مسح او بحث
27 Route::resource('/reservation', 'ReservationController');
28
29 -----end reservation pages-----
30

```

```

1  /*-----start menu pages-----*/
2
3 // وظيفته الذهاب الى لينك الموظف ثم تنفيذ الكود المطلوب سواء اضافة او تعديل او مسح او بحث
4 Route::resource('/menu-section','MenuSectionController');
5
6 // وظيفته الذهاب الى لينك الموظف ثم تنفيذ الكود المطلوب سواء اضافة او تعديل او مسح او بحث
7 Route::resource('/menu-items','MenuItemController');
8
9 Route::get('/item/{id}', 'MenuItemController@getitem');
10
11 /*-----end menu pages-----*/
12
13 /*-----start order pages-----*/
14
15 // وظيفته الذهاب الى لينك الموظف ثم تنفيذ الكود المطلوب سواء اضافة او تعديل او مسح او بحث
16 Route::resource('/order', 'OrderController');
17
18 /*-----end order pages-----*/
19
20 /*-----start bill pages-----*/
21
22 // وظيفته الذهاب الى لينك الموظف ثم تنفيذ الكود المطلوب سواء اضافة او تعديل او مسح او بحث
23 Route::resource('/bill', 'CheckOutController');
24
25 /*-----end bill pages-----*/
26

```

```

1  /*-----start Reports pages-----*/
2
3 // فى هذا الجزء يتم ربط صفحات التقارير و عمليات البحث بصفحات controllers الخاصة بهم
4
5 Route::get('/report', 'Emp_Report@index');
6 Route::post('/search_employee', 'Emp_Report@search_emp');
7
8 Route::get('/res_report','reservation_report@index');
9 Route::post('/search_reservation','reservation_report@search_res');
10
11 Route::get('/report_order','order_report@index');
12 Route::post('/search_order','order_report@search_order');
13
14 Route::get('/report_invoice','invoice_report@index');
15 Route::post('/search_invoice','invoice_report@search_invoice');
16
17 /*-----end Reports pages-----*/
18
19 // يقوم بربط صفحات login , registration بصفحات controllers الخاصة بهم
20 Auth::routes(['register' => true]);
21 // home // home يفتح صفحة login او registration بعد القيام بعملية
22 Route::get('/home', 'HomeController@index')->name('home');
23

```

: DB Migrations - ملفات*** جدول Users**

```

1 public function up()
2 {
3     Schema::create('users', function (Blueprint $table) {
4         $table->id();
5         $table->string('name');
6         $table->string('user_name')->unique();
7         $table->string('password');
8         $table->timestamps();
9     });
10    }
11

```

*** جدول Area Sections**

```

1 public function up()
2 {
3     Schema::create('area_sections', function (Blueprint $table) {
4         $table->id();
5         $table->string('section_name')->unique();
6         $table->timestamps();
7     });
8 }
9

```

*** جدول Reservation :**

```

1 public function up()
2 {
3     Schema::create('reservations', function (Blueprint $table) {
4         $table->id();
5         $table->string('cust_name');
6         $table->char('cust_phone',11);
7         $table->integer('count_person');
8         $table->date('reserve_date');
9         $table->date('customer_date');
10        $table->time('customer_time');
11        $table->unsignedBigInteger('section_name');
12        $table->string('user');
13        $table->foreign('section_name')->references('id')->on('area_sections')->onDelete('cascade')->onUpdate('cascade');
14        $table->timestamps();
15    });
16 }

```

*** جدول Tables :**

```

1 public function up()
2 {
3     Schema::create('tables', function (Blueprint $table) {
4         $table->id();
5         $table->Integer('table_number')->unique();
6         $table->unsignedBigInteger('area_id');
7         $table->foreign('area_id')->references('id')->on('area_sections')->onDelete('cascade');
8         $table->timestamps();
9     });
10 }
11

```

*** جدول :Menu Sections**

```

1 public function up()
2 {
3     Schema::create('menu_sections', function (Blueprint $table) {
4         $table->id();
5         $table->string('section_name')->unique();
6         $table->timestamps();
7     });
8 }
9

```

*** جدول :Menu Items**

```

1 public function up()
2 {
3     Schema::create('menu_items', function (Blueprint $table) {
4         $table->id();
5         $table->string('item_name')->unique();
6         $table->integer('price');
7         $table->unsignedBigInteger('section_id');
8         $table->foreign('section_id')->references('id')->on('menu_sections')->onDelete('cascade')->onUpdate('cascade');
9         $table->timestamps();
10    });
11 }
12

```

*** جدول :Orders**

```

1 public function up()
2 {
3     Schema::create('orders', function (Blueprint $table) {
4         $table->id();
5         $table->date('order_date');
6         $table->time('order_time');
7         $table->string('section_name');
8         $table->string('notes')->nullable();
9         $table->integer('sub_total');
10        $table->unsignedBigInteger('table_id');
11        $table->string('user');
12        $table->foreign('table_id')->references('id')->on('tables')->onDelete('cascade')->onUpdate('cascade');
13        $table->timestamps();
14    });
15 }

```

*** جدول :Order Items**

```

1 public function up()
2 {
3     Schema::create('order_items', function (Blueprint $table) {
4         $table->id();
5         $table->unsignedBigInteger('order_id');
6         $table->unsignedBigInteger('item_id');
7         $table->integer('quantity');
8         $table->integer('total');
9         $table->foreign('order_id')->references('id')->on('orders')->onDelete('cascade')->onUpdate('cascade');
10        $table->foreign('item_id')->references('id')->on('menu_items')->onDelete('cascade')->onUpdate('cascade');
11        $table->timestamps();
12    });
13 }

```

* جدول :Invoices

```
1 public function up()
2 {
3     Schema::create('check_outs', function (Blueprint $table) {
4         $table->id();
5         $table->date('invoice_date');
6         $table->time('invoice_time');
7         $table->double('service');
8         $table->double('tax');
9         $table->double('amount_total');
10        $table->unsignedBigInteger('order_id');
11        $table->string('table_id');
12        $table->string('user');
13        $table->foreign('order_id')->references('id')->on('orders')->onDelete('cascade')->onUpdate('cascade');
14        $table->timestamps();
15    });
16 }
```

* جدول :Employees

```
1 public function up()
2 {
3     Schema::create('employees', function (Blueprint $table) {
4         $table->id();
5         $table->string('name');
6         $table->string('phone');
7         $table->string('address');
8         $table->date('hire_date');
9         $table->integer('salary');
10        $table->string('job_type');
11        $table->string('gender');
12        $table->timestamps();
13    });
14 }
```

: ملفات - DB Models**: User model ***

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8
9 class User extends Authenticatable
10 {
11     use Notifiable;
12
13     protected $fillable = [
14         'name', 'user_name', 'password',
15     ];
16
17 }

```

: Area Section model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class area_sections extends Model
8 {
9     protected $fillable = [
10         'section_name',
11     ];
12 }
13

```

: Reservation model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class reservation extends Model
8 {
9     protected $fillable = [
10         'cust_name',
11         'cust_phone',
12         'count_person',
13         'reserve_date',
14         'customer_date',
15         'customer_time',
16         'section_name',
17         'user',
18     ];
19
20
21     public function section() // يتم استخدامها لظهور اسم قسم الطاولات عن طريق استدعاء جدول الاقسام
22     {
23         return $this->belongsTo('App\area_sections','section_name');
24     }
25 }
26

```

: Tables model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class table extends Model
8 {
9     protected $fillable = [
10         'table_number',
11         'area_id',
12     ];
13
14
15     public function sections() // يتم استخدامها لظهور اسم قسم الطاولات عن طريق استدعاء جدول الاقسام
16     {
17         return $this->belongsTo('App\area_sections','area_id');
18     }
19 }
20

```

: Menu Section model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class menu_section extends Model
8 {
9     protected $fillable = [
10         'section_name',
11     ];
12 }

```

: Menu Items model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class menu_item extends Model
8 {
9     protected $fillable = [
10         'item_name',
11         'price',
12         'section_id',
13     ];
14
15     public function section() // يتم استخدامها لظهور اسم قسم الطاولات عن طريق استدعاء جدول الاقسام
16     {
17         return $this->belongsTo('App\menu_section','section_id');
18     }
19
20     // public function orders()
21     // {
22     //     return $this->hasMany('App\order','order_item');
23     // }
24 }

```

: Order model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class order extends Model
8 {
9     protected $fillable = [
10         'order_date',
11         'order_time',
12         'section_name',
13         'notes',
14         'sub_total',
15         'table_id',
16         'user',
17     ];
18
19     public function section() // يتم استخدامها لظهور اسم قسم الطاولات عن طريق استدعاء جدول الاقسام
20     {
21         return $this->belongsTo('App\area_sections','section_name');
22     }
23
24
25     public function tnumber() // يتم استخدامها لظهور رقم الطاولة عن طريق استدعاء جدول الطاولات
26     {
27         return $this->belongsTo('App\table','table_id');
28     }
29
30 }
31

```

: Order Item model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class order_item extends Model
8 {
9     protected $table = 'order_items';
10
11     protected $fillable = ['order_id','item_id','quantity','total'];
12
13     public function item() // menu_item يتم استخدامها لظهور اسم العنصر عن طريق استدعاء جدول العناصر
14     {
15         return $this->belongsTo('App\menu_item','item_id');
16     }
17 }
18

```

: Invoice model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class check_out extends Model
8 {
9     protected $fillable = [
10         'invoice_date',
11         'invoice_time',
12         'service',
13         'tax',
14         'amount_total',
15         'order_id',
16         'table_id',
17         'user',
18     ];
19
20     public function item() // menu_item عن طريق استدعاء جدول
21     {
22         return $this->belongsTo('App\order', 'order_id');
23     }
24
25     public function tnumber() // رقم الطاولة عن طريق استدعاء جدول الطاولات
26     {
27         return $this->belongsTo('App\table', 'table_id');
28     }
29 }

```

: Employee model *

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class employee extends Model
8 {
9     protected $fillable = [
10         'name',
11         'phone',
12         'address',
13         'hire_date',
14         'salary',
15         'job_type',
16         'gender',
17     ];
18 }
19

```

- ملفات :views and controllers

:Login View *

```

1 <form method="POST" action="{{ route('login') }}>
2     @csrf
3         <div class="login-form">
4             <h1>login to yamifood system</h1>
5             <input placeholder="Enter UserName *" id="user_name" type="text"
6                 class="form-control @error('user_name') is-invalid @enderror"
7                 name="user_name" value="{{ old('user_name') }}" required autocomplete="off"
8             >
9             @error('user_name')
10                <span class="invalid-feedback" role="alert">
11                    <strong>{{ $message }}</strong>
12                </span>
13            @enderror
14
15             <input placeholder="Enter Password *" id="password" type="password"
16                 class="form-control @error('password') is-invalid @enderror"
17                 name="password" required autocomplete="current-password">
18
19             @error('password')
20                <span class="invalid-feedback" role="alert">
21                    <strong>{{ $message }}</strong>
22                </span>
23            @enderror
24             <input type="submit" value="Login"/>
25         </div>
26     </form>

```

:Login Function *

```

1 public function __construct()
2 {
3     $this->middleware('guest')->except('logout')
4 }
5
6 public function username()
7 {
8     return 'user_name';
9 }

```

:Registration View *

```

1  <form method="POST" action="{{ route('register') }}>
2      @csrf
3      <div class="signup-form">
4          <h1>signup to yamifood</h1>
5          <input placeholder="Enter Your Name *" id="name" type="text"
6              class="form-control @error('name') is-invalid @enderror"
7              name="name" value="{{ old('name') }}" required autocomplete="off">
8          @error('name')
9              <span class="invalid-feedback" role="alert">
10                 <strong>{{ $message }}</strong>
11             </span>
12         @enderror
13
14         <input placeholder="Enter UserName *" id="user_name" type="text"
15             class="form-control @error('user_name') is-invalid @enderror"
16             name="user_name" value="{{ old('user_name') }}" required autocomplete="off">
17         @error('user_name')
18             <span class="invalid-feedback" role="alert">
19                 <strong>{{ $message }}</strong>
20             </span>
21         @enderror
22
23         <input placeholder="Enter Your Password *" id="password" type="password"
24             class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="new-password"
25         @error('password')
26             <span class="invalid-feedback" role="alert">
27                 <strong>{{ $message }}</strong>
28             </span>
29         @enderror
30         <button type="submit" class="btn-sub">signup</button>
31         <button type="reset" class="btn-clear">clear</button>
32     </div>
33 </form>

```

:Registration Function *

```

1  public function __construct()
2  {
3      $this->middleware('guest');
4  }
5
6  protected function validator(array $data)
7  {
8      return Validator::make($data, [
9          'name' => ['required', 'string', 'max:255'],
10         'user_name' => ['required', 'string', 'max:255', 'unique:users'],
11         'password' => ['required', 'string', 'min:8', 'confirmed'],
12     ]);
13 }
14
15 protected function create(array $data)
16 {
17
18     return User::create([
19         'name' => $data['name'],
20         'user_name' => $data['user_name'],
21         'password' => Hash::make($data['password']),
22     ]);
23 }

```

:Save Area Section view *

```

1 <!-- تنفيذ كود الاضافة الخاص بالاقسام --!
2     <form action="{{ route('area-section.store') }}" method="post">
3         {{ csrf_field() }}
4
5             <div class="form-group">
6                 <label for="exampleInputEmail1">Area Name</label>
7                 <input type="text" class="form-control" id="section_name" name="section_name" autocomplete="off"
8             </div>
9
10            <div class="modal-footer">
11                <button type="submit" class="btn btn-success">Save</button>
12                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
13            </div>
14        </form>

```

:Save Area Section Function *

```

1 // عملية التحقق من البيانات وعملية اضافة الاقسام الخاصة بالمعطم
2 public function store(Request $request)
3 {
4     // التحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'section_name'      => 'required|unique:area_sections|max:255'
7     ], [
8         'section_name.required'    =>'Please Enter Area Section Name <br>',
9         'section_name.unique'      =>'This Area Section Name is Already Exist'
10    ]);
11
12    // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
13    if ($validator->fails()) {
14        return back()->with('toast_error', $validator->messages()->all())->withInput
15    ());
16
17    area_sections::create([ // الكود الخاص بعملية اضافة الاقسام الخاصة بالمعطم //
18
19        'section_name'=>$request->section_name
20    ]);
21
22    return redirect('area-section')->with('toast_success', 'area section Created!');
23 }

```

:Show Area Section view *

```

1 <div class="search-part">
2     <table>
3         <thead>
4             <tr>
5                 <th>ID</th>
6                 <th>area section</th>
7                 <th>delete</th>
8             </tr>
9         </thead>
10        <tbody>
11            <!-- استرجاع جميع بيانات الاقسام الموجودة -->
12            <?php $i=0; ?>
13            @foreach($area as $r)
14            <?php $i++ ?>
15            <tr>
16                <td>{{ $i }}</td>
17                <td>{{ $r->section_name }}</td>
18
19                <!-- اللينك الخامس بتشغيل شاشة مسح الاقسام الخاصة بالمطعم -->
20                <td><a href="#modaldemo9" data-toggle="modal" data-id="{{ $r->id }}" data-section_name="{{ $r->section_name }}"
21 e >" data-toggle="modal" href="#modaldemo9"><i class="fas fa-trash"></i></a>
22                </td>
23            </tr>
24            @endforeach
25        </tbody>
26    </table>
27 </div>

```

:Show Area Section Function *

```

1 // اول عملية يتم تنفيذها الى من
2 public function index()
3 {
4     // استرجاع جميع بيانات الاقسام الخاصة بالمطعم
5     $area = area_sections::all();
6     return view('area-section',compact('area'));
7 }

```

:Delete Area Section View *

```

1 <!-- area-section/destroy -->
2     <form action="area-section/destroy" method="post">
3         {{ method_field('delete') }}
4         {{ csrf_field() }}
5
6         <div class="form-group">
7             <p>Do you sure delete Area section</p>
8             <label for="exampleInputEmail1">Area Name</label>
9             <input type="hidden" name="id" id="id" value="">
10            <input type="text" class="form-control" id="section_name" name="section_name" autocomplete="off">
11        </div>
12
13        <div class="modal-footer">
14            <button type="submit" class="btn btn-success">Delete</button>
15            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16        </div>
17    </form>

```

```

1 <script>
2 // الكود الخاص بإظهار الاسم القسم في شاشة المسح
3 $('#modaldemo9').on('show.bs.modal', function(event) {
4     var button = $(event.relatedTarget)
5     var id = button.data('id')
6     var section_name = button.data('section_name')
7     var modal = $(this)
8     modal.find('.modal-body #id').val(id);
9     modal.find('.modal-body #section_name').val(section_name)
10    });
11
12 </script>

```

:Delete Area Section Function *

```

1 // الكود الخاص بعملية مسح الاقسام الخاصة بالمطعم
2 public function destroy(Request $request)
3 {
4     $id = $request->id;
5     area_sections::find($id)->delete();
6     return redirect('area-section')->with('toast_success', 'Area Section Deleted!');
7 }

```

:Save Reservation view *

```

1 // تنفذ لتشغيل صفحة إضافة الحجز //
2 public function create()
3 {
4     // عرض جميع أسماء اقسام الطاولات //
5     $area = area_sections::all();
6     // يتم تنفيذ هذه البيانات في شاشة الحجز //
7     return view('reservation',compact('area'));
8 }

```



```

1 <!-- route reservation.store --> تنفيذ كود اضافة الحجز
2     <form action="{{route('reservation.store')}}" method="post" class="reservation-form" autocomplete="off">
3         {{csrf_field()}}
4             <input type="text" name="customer_name" placeholder="customer name *"/>
5             <input type="text" name="customer_phone" placeholder="customer phone *"/>
6             <input type="number" name="customer_count" placeholder="count of person *"/>
7             <input type="date" name="customer_date" class="left-side-2"/>
8
9
10    <select list="area-place" name="section" id="area_number" placeholder="area section *" class="left-side-1">
11        <option value="" disabled selected>select section</option>
12        @foreach($area as $r)
13            <!-- عملية ظهور اسماء الاقسام ولوجود اكثر من قسم يتم استخدام كعملية تكرار --!
14            <!-- لعرض اسم القسم
15            <option value="{{ $r->id }}> {{ $r->section_name }} </option>
16            @endforeach
17        </select>
18
19        <input type="date" readonly value="{{date('Y-m-d')}}" name="reserve_date" />
20
21        <input type="time" name="customer_time" class="left-side-3" />
22
23        <button type="submit" class="btn-con-1">add</button>
24        <button type="reset" class="btn-con-2">clear</button>
25
26    </form>
27    <!--end reservation form-->
28    @include('sweetalert::alert')

```

Save Reservation Function *

```

1 // عملية التحقق من البيانات المدخلة وتنفيذ اضافة الحجز
2 public function store(Request $request)
3 {
4     // التحقق من البيانات
5     $validator = Validator::make($request->all(), [
6         'customer_name'      => 'required|min:8',
7         'customer_phone'     => 'required|numeric|digits:11|starts_with:0',
8         'customer_count'    => 'required|integer|min:1',
9         'customer_date'      => 'required|date|after_or_equal:today',
10        'section'           => 'required',
11    ],
12    [
13        //-----
14        'customer_name.required'   =>'Please Enter customer name <br>',
15        'customer_name.min'       =>'Error Entering Name This Name is Less Than 8 Letters <br>',
16        //-----
17        //-----
18        'customer_phone.required' =>'Please Enter customer Phone <br>',
19        'customer_phone.numeric'  =>'Error Entering Number Phone Please Enter Number Phone <br>',
20        'customer_phone.digits'   =>'Error Entering Number Phone This Number Phone is Less Than Or Greater Than 11 Number <br>'
21        'customer_phone.starts_with' =>'Error Entering Number Phone Please Enter Number Phone <br>',
22        //-----
23        //-----
24        //-----
25        'customer_count.required' =>'Please Enter customer count <br>',
26        'customer_count.integer'  =>'Error Entering customer count Please Enter customer count <br>',
27        'customer_count.min'      =>'Error Entering customer count Please Enter customer count <br>',
28        //-----
29        //-----
30        //-----
31        'customer_date.required'  =>'Please Enter Date Time <br>',
32        'customer_date.after_or_equal' =>'Error Entering Data Time Please Enter Date Today Or After Day <br>',
33        //-----
34        //-----
35        //-----
36        'section.required'        =>'Please Enter section name <br>',
37        //-----
38    ],
39);
40
41 // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات
42 if ($validator->fails()) {
43     return back()->with('toast_error', $validator->messages()->all())->withInput();
44 }
45
46 reservation::create() // الكود الخام بتنفيذ اضافة الحجز
47     'cust_name'=>$request->customer_name,
48     'cust_phone'=>$request->customer_phone,
49     'count_person'=>$request->customer_count,
50     'reserve_date'=>$request->reserve_date,
51     'customer_date'=>$request->customer_date,
52     'customer_time'=>$request->customer_time,
53     'section_name'=>$request->section,
54     'user'=>(Auth::user()->name), // شرح
55 ];
56
57 // ( الشهاب الي شاشة قائمة العجوزات مع تنفيذ اضافة الحجز مع ظهور رسالة (تم اضافة الحجز
58 return redirect('reservation')->with('toast_success','Reservation Created!');
59 }

```

:Show Reservation view *

```

1 <div class="search-part">
2   <table>
3     <thead>
4       <tr>
5         <th>ID</th>
6         <th>customer name</th>
7         <th>customer phone</th>
8         <th>count of person</th>
9         <th>area place</th>
10        <th>reserved date</th>
11        <th>customer date</th>
12        <th>customer date</th>
13        <th>user name</th>
14        <th>update</th>
15        <th>delete</th>
16      </tr>
17    </thead>
18    <tbody>
19      <!-- استدعاء جميع بيانات الحجوزات -->
20      <?php $i = 0; ?>
21      @foreach($reserve as $res)
22      <?php $i++ ?>
23      <tr>
24        <td>{{ $i }}</td>
25        <td>{{$res->cust_name}}</td>
26        <td>{{$res->cust_phone}}</td>
27        <td>{{$res->count_person}}</td>
28        <td>{{$res->section->section_name}}</td>
29        <td>{{$res->reserve_date}}</td>
30        <td>{{$res->customer_date}}</td>
31        <td>{{$res->customer_time}}</td>
32        <td>{{$res->user}}</td>
33        <td>-- التوink الخامس بارسال بيانات الحجز في شاشة تعديل الحجز -->
34        <a href="{{route('reservation.edit',$res['id'])}}><i class="fas fa-edit"></i></a></td>
35      <td>
36        <!-- لينك شاشة الخدف -->
37        <a href="#modaldemo9" data-toggle="modal" data-id="{{ $res->id }}" data-cust_name="{{$res->cust_name}}"
38          data-toggle="modal" class="del" href="#modaldemo9"><i class="fas fa-trash"></i></a>
39      </td>
40
41      </tr>
42    @endforeach
43  </tbody>
44 </table>
45 </div>

```

:Show Reservation Function *

```

1 // اول عملية يتم تنفيذها ال من
2 public function index(Request $request)
3 {
4   // عرض جميع اسماء اقسام الطاولات //
5   $reserve = area_sections::all();
6   // عرض جميع بيانات الحجوزات //
7   $reserve = reservation::all();
8
9   // يتم عرض هذه البيانات في شاشة قائمة الحجوزات //
10  return view('reservation-controls',compact('reserve'
11 )); }
12

```

:Edit Reservation Function *



```

1 // عرض بيانات الحجز في شاشة الخاصة بالتعديل على بيانات الحجز //
2 public function edit($id)
3 {
4     // عرض جميع اسماء اقسام الطاولات //
5     $area = area_sections::all();
6     // عرض بيانات الحجز من خلال رقم الحجز //
7     $res = reservation::find($id);
8     // الذهاب الى شاشة تعديل بيانات الحجز مع تنفيذ استرجاع بيانات الحجز من خلال رقم الحجز //
9     return view('/reservation-update',compact('res','area'));
10 }

```

:Update Reservation View *



```

1 <!-- action('EmployeeController@update', $id) -->
2 <form method="post" action="/reservation/update" class="reservation-update-form" autocomplete="off">
3 {{method_field('patch')}}
4 {{csrf_field()}}
5     <input type="hidden" value="{{ $res->id }}" name="res_id" />
6     <input type="text" value="{{ $res->cust_name }}" name="customer_name" placeholder="customer name *" />
7     <input type="text" value="{{ $res->cust_phone }}" name="customer_phone" placeholder="customer phone *" />
8     <input type="number" value="{{ $res->count_person }}" name="customer_count" placeholder="count of person *" />
9     <input type="date" value="{{ $res->customer_date }}" name="customer_date" class="left-side-2"/>
10    <select list="area-place" name="section" id="area_number" placeholder="area section *" class="left-side-1 "
11        onchange="" >
12        <option value="{{ $res->section_name }}" selected>{{ $res->section->section_name }}</option>
13        <!-- عملية ظهور اسماء الاقسام ولوجود اكثر من قسم يتم استخدام كعملية تكرار -->
14        @foreach($area as $r)
15            <option value="{{ $r->id }}> {{ $r->section_name }} </option>
16        @endforeach
17    </select>
18
19    <input type="" readonly value="{{ $res->reserve_date }}" name="reserve_date" />
20
21    <input type="time" name="customer_time" value="{{ $res->customer_time }}" placeholder="any notes" class="left-side-3" /
22 >
23    <button type="submit" class="btn-con-1">update</button>
24    <button type="reset" class="btn-con-2">clear</button>
25 </form>
26 <!--end reservation update form-->
27 @include('sweetalert::alert')

```

Update Reservation Function *

:Delete Reservation view *

```

1 <div class="modal-body">
2     <!-- تنفيذ كود الحذف الخاص بالحجز -->
3         <form action="reservation/destroy" method="post">
4             {{ method_field('delete') }}
5             {{ csrf_field() }}
6
7             <div class="form-group">
8                 <p>Do you sure delete reservation </p>
9                 <label for="exampleInputEmail1">Customer Name</label>
10                <input type="hidden" name="id" id="id" value="">
11                <input type="text" class="form-control" id="cust_name" name="cust_name" autocomplete="off">
12            </div>
13
14            <div class="modal-footer">
15                <button type="submit" class="btn btn-success">Delete</button>
16                <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
17            </div>
18        </form>
19    </div>
20    @include('sweetalert::alert')

```



```

1 <script>
2     // الكود الخاص بإظهار اسم الحجز في شاشة الحذف //
3     $('#modaldemo9').on('show.bs.modal', function(event) {
4         var button = $(event.relatedTarget)
5         var id = button.data('id')
6         var cust_name = button.data('cust_name')
7         var modal = $(this)
8         modal.find('.modal-body #id').val(id);
9         modal.find('.modal-body #cust_name').val(cust_name)
10    });
11
12 </script>

```

:Delete Reservation Function *

```

1 // الكود الخام بعملية المسح على بيانات الحجز
2 public function destroy(Request $request)
3 {
4     $id = reservation::findOrFail($request->id);
5     $id->delete();
6     return redirect('reservation')->with('toast_success','Delete successfully')
7 ); }
```

:Search Reservation View *

```

1 <form method="get" action="/search_res">
2     {{csrf_field()}}
3     <input class="txt_search" autocomplete="off" name="search" type="search" placeholder="search for reservation" />
4     <input type="date" name="search_date" class="txt_search1"/>
5     <button type="submit" class="btn_search">search</button>
6 </form>
```

:Search Reservation Function *

```

1 // يتم من خلالها عمل بحث
2 public function search_res(Request $request)
3 {
4     if($search = $request->get('search')){
5
6         $search = $request->get('search');
7         $reserve = reservation::select('*')->where('cust_name' , $search)
8             ->orwhere('cust_phone' , $search)->get();
9
10        return view('reservation-controls',compact('reserve'));
11    }
12
13    if($search_date = $request->search_date){
14
15        $search_date = date($request->search_date);
16        $reserve = reservation::select('*')->where('customer_date' , $search_date)->get();
17
18        return view('reservation-controls',compact('reserve'));
19    }
20
21 }
```

:Save Tables view *

```

1 <!-- تنفيذ كود الاضافة الخاص بالطاولات -->
2     <form action="{{ route('tables.store') }}" method="post">
3         {{ csrf_field() }}
4         <div class="modal-body">
5             <div class="form-group">
6                 <label for="exampleInputEmail1">Table Number</label>
7                 <input type="number" class="form-control" id="table_number" name="table_number" autocomplete="off"
8             </div>
9
10            <label class="my-1 mr-2" for="inlineFormCustomSelectPref">Section Name</label>
11            <select name="section_id" id="section_id" class="form-control">
12                <option value="" selected disabled>Select Section</option>
13                @foreach ($sections as $section)
14                    <option value="{{ $section->id }}>{{ $section->section_name }}</option>
15                @endforeach
16            </select>
17
18        </div>
19        <div class="modal-footer">
20            <button type="submit" class="btn btn-success">save</button>
21            <button type="button" class="btn btn-secondary" data-dismiss="modal">close</button>
22        </div>
23    </form>

```

:Save Tables Function *

```

1 // عملية التحقق من البيانات وعملية اضافة الطاولات الخاصة بالمطعم //
2 public function store(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'table_number'          => 'required|unique:tables',
7         'section_id'           => 'required'
8     , [
9
10         'table_number.required'   =>'Please Enter Table Number <br>',
11         'table_number.starts_with' =>'Error Entering Number Please Enter Table Number <br>'
12     , 'table_number.unique'      =>'This Table Number is Already Exist',
13         'section_id.required'    =>'Please Enter Section Name'
14     ]);
15
16
17     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
18     if ($validator->fails()) {
19         return back()->with('toast_error', $validator->messages()->all()->withInput());
20     }
21
22     table::create([ // الكود الخاص بعملية اضافة الطاولات الخاصة بالمطعم
23
24         'table_number'=>$request->table_number,
25         'area_id'=>$request->section_id,
26     ]);
27
28     return redirect('tables')->with('toast_success', 'Add Table Created!');
29 }

```

:Show Tables view *

```

1 <div class="search-part">
2     <table>
3         <thead>
4             <tr>
5                 <th>table number</th>
6                 <th>area section</th>
7                 <th>delete</th>
8             </tr>
9         </thead>
10        <tbody>!-- استرجاع جميع بيانات الطاولات الموجودة -->
11        <tr>
12            @foreach($table as $t)
13            <td>{{ $t->table_number }}</td>
14            <td>{{ $t->sections->section_name }}</td>
15
16            <td>!-- اللينك الخاص بتشغيل شاشة مسح الطاولة الخاصة بالمطعم -->
17            <a href="#modaldemo9" data-toggle="modal" data-id="{{ $t->id }}" data-table_number="{{ $t->table_number }}"
18 r }}">
19             data-toggle="modal" href="#modaldemo9"><i class="fas fa-trash"></i></a>
20             </td>
21         </tr>
22     @endforeach
23     </tbody>
24 </table>

```

:Show Tables Function *

```

1 // اول عملية يتم تنفيذها
2 public function index()
3 {
4     // استرجاع جميع بيانات عدد الطاولات الخاصة بالمطعم //
5     $sections = area_sections::all();
6     $table = table::all();
7     return view('tables-controls',compact('sections','table'
8 ));}
9

```

:Delete Tables View *

```

1 <!-- تنفيذ كود الافاقنة الخام بالطابولات -->
2     <form action="tables/destroy" method="post">
3         {{ method_field('delete') }}
4         {{ csrf_field() }}
5
6         <div class="form-group">
7             <p>Do you sure delete Table Number</p>
8             <label for="exampleInputEmail1">Table Number</label>
9             <input type="hidden" name="id" id="id" value="">
10            <input type="number" class="form-control" id="table_number" name="table_number" autocomplete="off">
11        </div>
12
13        <div class="modal-footer">
14            <button type="submit" class="btn btn-success">Delete</button>
15            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16        </div>
17    </form>

```

```

1 <script>
2 // الكود الخام بإظهار الاسم القسم فى شاشة المسح
3 $('#modaldemo9').on('show.bs.modal', function(event) {
4     var button = $(event.relatedTarget)
5     var id = button.data('id')
6     var table_number = button.data('table_number')
7     var modal = $(this)
8     modal.find('.modal-body #id').val(id);
9     modal.find('.modal-body #table_number').val(table_number
10 ); })
11 </script>
12

```

:Delete Tables Function *

```

1 // الكود الخام بعملية مسح الطابولات الخاصة بالمطعم
2 public function destroy(Request $request)
3 {
4     $id = $request->id;
5     table::find($id)->delete();
6     return redirect('tables')->with('toast_success', 'Table Number Deleted!');
7 }

```

:Save Menu Section view *

```

1 <!-- تنفيذ كود الاضافة الخام بالاقسام قائمة الطعام --!>
2     <form action="{{ route('menu-section.store') }}" method="post">
3         {{ csrf_field() }}
4
5         <div class="form-group">
6             <label for="exampleInputEmail1">Section Name</label>
7             <input type="text" class="form-control" id="section_name" name="section_name" autocomplete="off"
8         </div>
9
10        <div class="modal-footer">
11            <button type="submit" class="btn btn-success">Save</button>
12            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
13        </div>
14    </form>

```

:Save Menu Section Function *

```

1 // عملية التحقق من البيانات وعملية اضافة القسم الخامة بقائمة الطعام
2 public function store(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'section_name'          => 'required|unique:menu_sections|max:255'
7     ], [
8
9         'section_name.required' =>'Please Enter Section Name <br>',
10        'section_name.unique'   =>'This Section Name is Already Exist'
11    ]);
12
13     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
14     if ($validator->fails()) {
15         return back()->with('toast_error', $validator->messages()->all())->withInput();
16     }
17
18     $section = new menu_section(); // الكود الخام بعملية اضافة القسم الخامة بقائمة الطعام
19     $section->section_name = $request->input('section_name');
20     $section->save();
21
22     return redirect('menu-section')->with('toast_success', 'Section Created!'); // تم اضافة القسم
23 } // الذهاب الى شاشة عرض الاقسام مع ظهور رسالة (تم اضافة القسم
24 }

```

: Menu Section Main Screen view *

```

1 <form class="menu-section-a">
2     <!-- عملية ظهور اسماء الاقسام ولو جود اكثر من قسم يتم استخدام foreach كعملية تكرار --!
3     @foreach($data as $d)
4         <div class="section-hot-drinks">
5             <!-- خامة برقم القسم -->
6             <!-- اللينك الخاص بظهور العناصر الخاصة بالقسم -->
7             <a class="mm" value="{{ $d->id }}" href="/item/{{ $d->id }}/{{ $d->section_name }}></a> <!-- لعرض اسم القسم -->
8         </div>
9     @endforeach
10    </form>
11

```

: Menu Section Main Screen Function *

```

1 // تنفذ لتشغيل صفحة الاقسام
2 public function create()
3 {
4     // عرض جميع اقسام الالكلات والمشروبات واي قسم اخ
5     // ر
6     $data = menu_section::all();
7
8     // يتم عرض هذه الاقسام في شاشة الاقسام
9     return view('menu-sections', compact('data', $data));
10

```

:Show Menu Section view *

```

1 <div class="search-part">
2     <table>
3         <thead>
4             <tr>
5                 <th>ID</th>
6                 <th>section name</th>
7                 <th>delete</th>
8             </tr>
9         </thead>
10        <tbody>
11            <!-- عرض جميع بيانات الاقسام الخامة بقائمة الطعام -->
12            <?php $i = 0; ?>
13            @foreach($menu as $m)
14                <?php $i++ ?>
15                <tr>
16                    <td>{{$i}}</td>
17                    <td>{{$m->section_name}}</td>
18
19                    <!-- #modaldemo8 -->
20                    <td><a href="#modaldemo9" data-toggle="modal" data-id="{{ $m->id }}" data-section_name="{{ $m->section_name }}"
21 e }}>
22                         data-toggle="modal" href="#modaldemo9"><i class="fas fa-trash"></i></a>
23                     </td>
24                 </tr>
25             @endforeach
26         </tbody>
27     </table>

```

:Show Menu Section Function *

```

1 // Route يتم تنفيذها من ال // 
2 public function index()
3 {
4     // عرض جميع اقسام الالектات والمشروبات واي قسم اخر //
5     $menu = menu_section::all();
6     // يتم استرجاع هذه البيانات في شاشة قائمة الاقسام
7     // return view('menu-section-control',compact('menu'));
8 }

```

:Delete Menu Section View *

```

1  <!-- تفاصيل كود الحذف الخاص بالقسم -->
2      <form action="menu-section/destroy" method="post">
3          {{ method_field('delete') }}
4          {{ csrf_field() }}

5          <div class="form-group">
6              <p>Do you sure delete section</p>
7              <label for="exampleInputEmail1">Section Name</label>
8              <input type="hidden" name="id" id="id" value="">
9              <input type="text" class="form-control" id="section_name" name="section_name" autocomplete="off">
10         </div>

11     <div class="modal-footer">
12         <button type="submit" class="btn btn-success">Delete</button>
13         <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
14     </div>
15 </form>
16
17

```

```

1  <script>
2      // الكود الخاص بإظهار اسم القسم في شاشة الحذف //
3      $('#modaldemo9').on('show.bs.modal', function(event) {
4          var button = $(event.relatedTarget)
5          var id = button.data('id')
6          var section_name = button.data('section_name')
7          var modal = $(this)
8          modal.find('.modal-body #id').val(id);
9          modal.find('.modal-body #section_name').val(section_name
10     });
11 </script>

```

:Delete Menu Section Function *

```

1  // الكود الخاص بعملية مسح الأقسام الخاصة بقائمة الطعام //
2  public function destroy(Request $request)
3  {
4      $id = $request->id;
5      menu_section::find($id)->delete();
6      return redirect('menu-section')->with('toast_success', 'Section Deleted!');
7  }

```

:Save Menu Item view *

```

1 <!-- تنفيذ كود الاضافة الخاص بالعنصر قائمة الطعام -->
2 <form action="{{ route('menu-items.store') }}" method="post">
3     {{ csrf_field() }}
4     <div class="modal-body">
5         <div class="form-group">
6             <label for="exampleInputEmail1">Item Name</label>
7             <input type="text" class="form-control" id="item_name" name="item_name" autocomplete="off" >
8         </div>
9
10        <div class="form-group">
11            <label for="exampleInputEmail1">Price</label>
12            <input type="number" class="form-control" id="price" name="price" autocomplete="off" >
13        </div>
14
15        <label class="my-1 mr-2" for="inlineFormCustomSelectPref">Section Name</label>
16        <select name="section_id" id="section_id" class="form-control">
17            <option value="" selected disabled>Select Section</option>
18            <!-- عملية عرض اسماء الاقسام ولو جود اكثر من قسم يتم استخدام كعملية تكرار -->
19            @foreach ($sections as $section)
20                <option value="{{ $section->id }}>{{ $section->section_name }}</option> <!-- لعرض اسم القسم -->
21            @endforeach
22        </select>
23
24    </div>
25    <div class="modal-footer">
26        <button type="submit" class="btn btn-success">save</button>
27        <button type="button" class="btn btn-secondary" data-dismiss="modal">close</button>
28    </div>
29 </form>

```

:Save Menu Item Function *

```

1 // عملية التحقق من البيانات وعملية اضافة العناصر الخاصة بقائمة الطعام //
2 public function store(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'item_name' => 'required|unique:menu_items|max:255',
7         'price' => 'required|integer|min:5',
8         'section_id' => 'required'
9     ], [
10         'item_name.required' =>'Please Enter Item Name <br>',
11         'item_name.unique' =>'This Item Name is Already Exist <br>',
12         'price.required' =>'Please Enter price <br>',
13         'price.integer' =>'Error Entering price Please Enter price <br>',
14         'price.min' =>'Error Entering price This Price is less Than 5 LE <br>',
15         'section_id.required' =>'Please Enter Section Name <br>',
16     ]);
17
18     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
19     if ($validator->fails()) {
20         return back()->with('toast_error', $validator->messages()->all())->withInput();
21     }
22
23     // الكود الخاص بعملية اضافة العناصر بالاقسام الخاصة بقائمة الطعام //
24     menu_item::create([
25
26         'item_name'=> $request->item_name,
27         'price'=> $request->price,
28         'section_id'=> $request->section_id,
29     ]);
30     // (الذهاب الى شاشة عرض قائمة العناصر بالاقسام مع ظهور رسالة (تم اضافة العنصر //
31     return redirect('menu-items')->with('toast_success', 'item Created!');
32 }
33

```

: Menu Item Main Screen view *



```

1 <form class="main-div">
2     <!-- عميلية ظهور اسماء العناصر الخاصة بالقسم ولوجود اكثر من عنصر يتم استخدام foreach كعملية تكرار --!
3 > @foreach($data as $d)
4     <div>
5         <!-- لعرض اسم العنصر -->
6         <a class="mm" value="{{ $d->id }}>{{ $d->item_name }}</a>
7     </div>
8 @endforeach
9 </form>

```

: Menu Item Main Screen Function *



```

1 // تنفذ لعرض العناصر الخاصة بقسم الطعام في شاشة عرض العناصر
2 public function getitem($id)
3 {
4     // ويكون عرض العناصر من خلال استدعاء رقم القسم
5     $data = menu_item::where('section_id',$id)->get();
6     // الذهاب لشاشة العناصر مع عرضها
7     return view('/item',compact('data',$data));
8 }

```

:Show Menu Item view *

```

1 <div class="search-part">
2     <table>
3         <thead>
4             <tr>
5                 <th>item name</th>
6                 <th>price</th>
7                 <th>section name</th>
8                 <th>update</th>
9                 <th>delete</th>
10            </tr>
11        </thead>
12        <tbody>
13            <!-- عرض جميع بيانات العناصر الخاصة بقائمة الطعام -->
14            @foreach($items as $item)
15                <tr>
16                    <td>{{ $item->item_name }}</td>
17                    <td>{{ $item->price }}</td>
18                    <td>{{ $item->section->section_name }}</td>
19
20                    <!-- #modaldemo8 -->
21                    <td><a href="#modaldemo12" data-item_name="{{ $item->item_name }}" data-item_id="{{ $item->id }}"
22                         data-price="{{ $item->price }}"
23                         data-section_name="{{ $item->section->section_name }}" data-toggle="modal"><i class="fas fa-edit"></i></a>
24                    </td>
25
26                    <!-- #modaldemo8 -->
27                    <td><a class="del" href="#modaldemo9" data-toggle="modal" data-id="{{ $item->id }}" data-item_name="{{ $item->item_name }}"
28                         data-toggle="modal" href="#modaldemo9"><i class="fas fa-trash"></i></a>
29                    </td>
30
31                @endforeach
32            </tbody>
33        </table>
34    </div>

```

:Show Menu Item Function *

```

1 // اول عملية يتم تنفيذها من ال
2 public function index()
3 {
4     // عرض جميع اقسام الالبات والمشروبات واي قسم اخر //
5     $sections = menu_section::all();
6     // عرض جميع العناصر اقسام الالبات والمشروبات //
7     $items = menu_item::all();
8     // يتم استرجاع هذه البيانات في شاشة قائمة العناصر الخاصة بالاقسام //
9     return view('menu-items-controls',compact('items','sections'));
10 }

```

:Update Menu Item view *

```

1  -- تنفيذ كود تعديل العنصر الخاص بقائمة الطعام ---
2      <form action="menu-items/update" method="post">
3          {{ method_field('patch') }}
4          {{ csrf_field() }}
5          <div class="modal-body">
6              <input type="hidden" class="form-control" name="item_id" id="item_id" value="">
7              <div class="form-group">
8                  <label for="exampleInputEmail1">Item Name</label>
9                  <input type="text" class="form-control" id="item_name" name="item_name" autocomplete="off" >
10             <label for="exampleInputEmail1">Price</label>
11             <input type="number" class="form-control" id="price" name="price" autocomplete="off" >
12         </div>
13
14
15     <label class="my-1 mr-2" for="inlineFormCustomSelectPref">Section Name</label>
16     <select name="section_name" id="section_name" class="form-control">
17         <!-- عملية عرض أسماء الأقسام ولوجود أكثر من قسم يتم استخدام كعملية تكرار --!
18         @foreach ($sections as $section)
19             <option>{{ $section->section_name }}</option> <!-- لعرض اسم القسم --
20         @endforeach
21     </select>
22
23
24     <div class="modal-footer">
25         <button type="submit" class="btn btn-success">update</button>
26         <button type="button" class="btn btn-secondary" data-dismiss="modal">close</button>
27     </div>
28 </form>

```



الكود الخاص بإظهار اسم العنصر والسعر واسم القسم الخاص به في شاشة التعدي //

```

1  // الكود الخاص بإظهار اسم العنصر والسعر واسم القسم الخاص به في شاشة التعدي //
2  // $('#modaldemo12').on('show.bs.modal', function(event) {
3      var button = $(event.relatedTarget)
4      var item_name = button.data('item_name')
5      var item_id = button.data('item_id')
6      var price = button.data('price')
7      var section_name = button.data('section_name')
8      var modal = $(this)
9      modal.find('.modal-body #item_name').val(item_name);
10     modal.find('.modal-body #price').val(price);
11     modal.find('.modal-body #section_name').val(section_name);
12     modal.find('.modal-body #item_id').val(item_id);
13 })

```

:Update Menu Item Function *



```

1 // عملية التحقق من البيانات وعملية تعديل العنصر الخامة بقائمة الطعام //
2 public function update(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'item_name' => 'required',
7         'price'      => 'required|integer|min:5',
8     ], [
9         'item_name.required'    =>'Please Enter Item Name <br>',
10        'price.required'       =>'Please Enter price <br>',
11        'price.integer'        =>'Error Entering price Please Enter price <br>',
12        'price.min'            =>'Error Entering price This Price is less Than 5 LE <br>',
13    ]);
14
15     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
16     if ($validator->fails()) {
17         return back()->with('toast_error', $validator->messages()->all())->withInput();
18     }
19
20     // استدعاء رقم القسم من جدول ال menu_section وذلك لربطها مع رقم العنصر في جدول //
21     $section_id = menu_section::where('section_name',$request->section_name)->first()->id;
22
23     // الكود الخاص بعملية تعديل العنصر بالقسم الخامة بقائمة الطعام من خلال رقم العنصر //
24     $item_id = menu_item::findOrFail($request->item_id);
25     $item_id->update([
26
27         'item_name'=> $request->item_name,
28         'price'=> $request->price,
29         'section_id'=> $section_id
30
31     ]);
32
33     // (الذهاب الى شاشة عرض قائمة العناصر بالاقسام مع ظهور رسالة (تم تعديل العنصر //
34     return redirect('menu-items')->with('toast_success','item Updated!'));
35 }
36

```

:Delete Menu Item view *

```

1  <!-- menu-section/destroy -->
2      <form action="menu-items/destroy" method="post">
3          {{ method_field('delete') }}
4          {{ csrf_field() }}
5
6          <div class="form-group">
7              <p>Do you sure delete Item </p>
8              <label for="exampleInputEmail1">Item Name</label>
9              <input type="hidden" name="id" id="id" value="">
10             <input type="text" class="form-control" id="item_name" name="item_name" autocomplete="off">
11         </div>
12
13         <div class="modal-footer">
14             <button type="submit" class="btn btn-success">Delete</button>
15             <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16         </div>
17     </form>

```



```

1  // الكود الخاص بإظهار اسم العنصر في شاشة الحذف
2  $('#modaldemo9').on('show.bs.modal', function(event) {
3      var button = $(event.relatedTarget)
4      var id = button.data('id')
5      var item_name = button.data('item_name')
6      var modal = $(this)
7      modal.find('.modal-body #id').val(id);
8      modal.find('.modal-body #item_name').val(item_name)
9  })

```

:Delete Menu Item Function *



```

1 // الكود الخاص بعملية مسح العنصر //
2 public function destroy(Request $request)
3 {
4     $id = $request->id;
5     menu_item::find($id)->delete();
6     return redirect('menu-items')->with('toast_success', 'Item Deleted!');
7 }

```

:Create Order Function *



```

1 // تنفيذ لتشغيل صفحة إضافة الطلب مع البيانات الجداول التالية //
2 public function create()
3 {
4     // عرض جميع أسماء اقسام الطاولات //
5     $area = area_sections::all();
6     // عرض جميع ارقام الطاولات //
7     $allTable = table::all();
8     // عرض جميع اقسام قائمة الطعام //
9     $menu = menu_section::all();
10    // عرض جميع اسماء عناصر قائمة الطعام //
11    $items = menu_item::all();
12
13    // يتم تنفيذ هذه البيانات في شاشة الطلب //
14    return view('order', compact('area', 'allTable', 'menu', 'items'
15 ));}
16

```

:Save Order view *

```

1  <!-- route order.store --> تنتهي كود اضافة الطلب
2  <form action="{{route('order.store')}}" method="post">
3  {{csrf_field()}}
4  <!--start order form-->
5
6  <div class="order-form">
7      <select list="area-place" name="section_name" id="area_number" placeholder="area section *" onchange="" -->
8          <option value="" disabled selected>select section</option>
9          <!-- عملية ظهور اسماء الاقسام ولو وجود اكثر من قسم يتم استخدام كعمليه تكرار -->
10         @foreach($area as $r)
11             <!-- لعرض اسم القسم -->
12             <option value="{{ $r->id }}> {{ $r->section_name }} </option>
13         @endforeach
14     </select>
15
16     <select list="table-id" name="table_id" id="clear_table" placeholder="table number *>
17         <option value="" disabled selected>select table</option>
18         <!-- عملية ظهور ارقام الطاولات ولو وجود اكثر من طاولة يتم استخدام كعمليه تكرار -->
19         @foreach($allTable as $item)
20             <!-- لعرض رقم الطاولة -->
21             <option value="{{ $item->id }}> {{ $item->area_id }} hide_table</option> {{ $item->table_number }} </option>
22         @endforeach
23     </select>
24
25
26     <select list="menu-section" id="item_name" placeholder="menu section *" onchange="" -->
27         <option value="" disabled selected>select menu</option>
28         <!-- عملية ظهور اسماء الاقسام ولو وجود اكثر من قسم يتم استخدام كعمليه تكرار -->
29         @foreach($menu as $m)
30             <!-- لعرض اسم القسم -->
31             <option value="{{ $m->id }}> {{ $m->section_name }} </option>
32         @endforeach
33     </select>
34
35     <!-- عملية استدعاء سعر العنصر ولو وجود اكثر من سعر عنصر يتم استخدام كعمليه تكرار -->
36     @foreach($items as $item)
37         <!-- لعرض سعر العنصر -->
38         <input class="{{ $item->item_name }}" type="hidden" value="{{ $item->price }}>
39     @endforeach
40
41     <!-- عملية استدعاء رقم العنصر ولو وجود اكثر من رقم عنصر يتم استخدام كعمليه تكرار -->
42     @foreach($items as $item)
43         <!-- لعرض رقم العنصر -->
44         <input class="id_{{ $item->item_name }}" type="hidden" value="{{ $item->id }}>
45     @endforeach
46
47     <select list="item-id" name="item_id" id="clear_item" placeholder="menu item *>
48         <option value="" disabled selected>select menu item</option>
49         <!-- عملية ظهور اسماء العناصر الخاصة بالقسم ولو وجود اكثر من عنصر يتم استخدام كعمليه تكرار -->
50         @foreach($items as $item)
51             <!-- لعرض اسم العنصر -->
52             <option value="{{ $item->item_name }}> {{ $item->section_id }} hide_table</option> {{ $item->item_name }} </option>
53         @endforeach
54     </select>
55
56
57     <input type="date" name="oreder_date" readonly value="{{date('Y-m-d')}}" class="right-side-1" />
58     <input type="time" name="oreder_time" readonly value="{{date('H:i:s')}}" class="right-side-2" />
59
60     <input type="text" name="notes" placeholder="any notes" class="right-side-4" />
61     <input type="text" name="sub_total" class="sub_total right-side-3" placeholder="sub-total" class="right-side-3" readonly />
62
63     <button type="submit">save</button>
64 </div>
65 <!--end order form-->
66
67 <!--start items form-->
68 <div class="items-form">
69     <table>
70         <thead>
71             <tr>
72                 <th>item name</th>
73                 <th>price</th>
74                 <th>quantity</th>
75                 <th>total</th>
76                 <th>ignore</th>
77             </tr>
78         </thead>
79         <tbody>
80             <!-- يتم هنا اضافة العنصر المختار -->
81         </tbody>
82     </table>
83 </div>
84 </form>
85 <!--end items form-->
```

```

1 <script>
2 // تنفذ عند اختيار اسم قسم طاولة يتم ظهور ارقام الطاولات الخاصة بالقسم
3 $(document).ready(function() {
4     $('.hide_table').hide();
5     $('#area_number').on('change', function() {
6         $area_id = $(this).val();
7
8         $('#clear_table').val('');
9         $('.hide_table').hide();
10        $('.table_-' + $area_id).show();
11    });
12
13 // تنفذ عند اختيار اسم قسم بقائمة الطعام يتم ظهور العنابر الخاصة بالقسم
14 $('.hide_table').hide();
15 $('#item_name').on('change', function() {
16     $section_id = $(this).val();
17
18     $('#clear_item').val('');
19     $('.hide_table').hide();
20     $('.table_-' + $section_id).show();
21 });
22 });
23
24 // الجزء الخاص بإضافة العنابر المختار وذلك من خلال اختيار اسم القسم من قائمة الطعام
25 // ثم اختيار العنابر وبعد ذلك يضاف إلى جدول الخاص بالطلب
26
27 $i = 0;
28 $('#clear_item').change(function() {
29     $i++;
30     $item_name = $(this).val();
31     $('option[value="" + $item_name + ""]').attr('disabled','disabled');
32     $price = $('input[class="" + $item_name + ""]').val();
33     $id = $('input[class="id_" + $item_name + ""]').val();
34
35     $('tbody').append('<tr>' +
36         '<td>' + $item_name + '<input type="hidden" name="item_id[]" value="' + $id + '"></td>' +
37         '<td>' + $price + '<input type="hidden" name="price[]" value="" + $price + "></td>' +
38         '<td><input class="quantity" type="number" name="quantity[]" value="0" data-price="" + $price + "" data-total="" + $i + "" min="1"></td>' +
39         '<td><input class="total num" value="" + $i + "" name="total[]" type="number" readonly value="" ></td>' +
40
41         '<td>' +
42         '<a class="btn btn-danger delRow text-white "><i class="fas fa-trash"></i><span class="name_item_hide">' + $item_name + '</span></a>' +
43         '</td>' +
44         '</tr>';
45 );
46
47 // الجزء الخاص بإختفاء اسم العنابر المختار من سلة حذف الطلب
48
49 // الجزء الخاص عند تحديد الكمية بقرب الكمية في السعر لحساب إجمالي الطلب الواحد
50 // وفي حالة وجود أكثر من طلب بجمع إجمالي كل طلب لحساب إجمالي الطلب بالكامل
51 // $.quantity'.on('change', function() {
52
53     $quan = $(this).val();
54     $num = $(this).data('total');
55     $price = $(this).data('price');
56     $total = $quan * $price;
57     $('.num' + $num).val($total);
58     var sum = 0;
59     $(".total").each(function() {
60         sum += +$(this).val();
61     });
62     $(".sub_total").val(sum);
63 });
64
65 // الجزء الخاص في حالة حذف عنصر وفي حالة حذفه يتم ظهور العنابر الخاصة به
66 // كما يتم طرح إجمالي هذا الطلب المحذوف من إجمالي الطلب بالكامل
67 // $.delRow'.on('click', function() {
68     $(this).parent().parent().remove(); // حذف العنابر
69
70     $('#clear_item').val('');
71     $('#item_name').val('');
72     // كما يتم طرح إجمالي هذا الطلب المحذوف من إجمالي الطلب بالكامل
73     var sum = 0;
74     $(".total").each(function() {
75         sum -= -$(this).val();
76     });
77     $(".sub_total").val(sum);
78
79     $span= $(this).text(); // ظهور العنابر الخاصة به
80     $('option[value="" + $span + ""]').removeAttr('disabled');
81
82 });
83 });
84 </script>

```

:Save Order Function *

```

1 // عملية التحقق من البيانات المدخلة وتنفيذ اضافة الطلب //
2 public function store(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'section_name' => 'required',
7         'table_id'      => 'required',
8         'item_id'       => 'required',
9         'sub_total'     => 'required',
10    ],
11    [
12        'section_name.required' => 'Please Enter Section Name <br>',
13        'table_id.required'      => 'Please Enter Table number <br>',
14        'item_id.required'       => 'Please Enter Item Name <br>',
15        'sub_total.required'     => 'sub total is field <br>',
16    ]);
17
18    // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
19    if ($validator->fails()) {
20        return back()->with('toast_error', $validator->messages()->all())->withInput();
21    }
22
23    $dateNow = date('Y-m-d');
24    $timeNow = date('H:i:s');
25
26    // الكود الخامس بتنفيذ اضافة الطلب الى جدول orders
27    $id = DB::table('orders')->insertGetId([
28
29        'order_date' => $dateNow,
30        'order_time' => $timeNow,
31        'section_name' => $request->section_name,
32        'notes' => $request->notes,
33        'sub_total' => $request->sub_total,
34        'table_id' => $request->table_id,
35        'user' => (Auth::user()->name)
36    ]);
37
38    // الكود الخامس بتنفيذ اضافة عناصر الطلب الى جدول order_item
39    for ($i = 0; $i < count($request->item_id); $i++) {
40        order_item::create([
41            'order_id' => $id,
42            'item_id' => $request->item_id[$i],
43            'quantity' => $request->quantity[$i],
44            'total' => $request->total[$i],
45        ]);
46    }
47    // return redirect('order')->with('toast_success', 'Order Created!');
48 }

```

الذهاب الى شاشة قائمة الطلبات مع تنفيذ اضافة الطلب مع ظهور رسالة (تم اضافة الطلب)

:Show Order View *

```

1  <!--start order controls form-->
2  <div class="order-controls">
3
4      <div class="search-part">
5          <table>
6              <thead>
7                  <tr>
8                      <th>ID</th>
9                      <th>section name</th>
10                     <th>table number</th>
11                     <th>date</th>
12                     <th>time</th>
13                     <th>user name</th>
14                     <th>update</th>
15                     <th>delete</th>
16                 </tr>
17             </thead>
18             <tbody>
19                 <!-- استدعاء جميع بيانات الطلبات -->
20                 @foreach($orders as $order)
21                     <tr>
22                         <td>{{$order->id}}</td>
23                         <td>{{$order->section->section_name}}</td>
24                         <!-- اللينك الخام بارسال بيانات الطلب في شاشة الفاتورة -->
25                         <td><a class="checkout" href="{{route('bill.edit',$order['id'])}}>{{$order->tnumber->table_number}}</a></td>
26                         <td>{{$order->order_date}}</td>
27                         <td>{{$order->order_time}}</td>
28                         <td>{{$order->user}}</td>
29
30                         <!-- اللينك الخاص بارسال بيانات الطلب في شاشة تديل الطلب -->
31                         <td><a href="{{route('order.edit',$order['id'])}}><i class="fas fa-edit"></i></a></td>
32
33                         <!-- لينك شاشة الحذف -->
34                         <td><a class="del" href="#modaldemo9" data-toggle="modal" data-id="{{$order->id}}" data-table_number="{{$order->tnumber->table_number}}><i class="fas fa-trash"></i></a>
35                     </td>
36                 </tr>
37             @endforeach
38         </tbody>
39     </table>
40
41 </div>
42

```

:Show Order Function *

```

1 // اول عملية يتم تنفيذها ال من
2 public function index()
3 {
4     // عرض جميع اسماء اقسام الولايات //
5     $area = area_sections::all();
6     // عرض جميع ارقام الولايات //
7     $allTable = table::all();
8     // عرض جميع بيانات الطلبات //
9     $orders = order::all();
10
11    // يتم عرض هذه البيانات في شاشة قائمة الطلبات //
12    return view('order-controls', compact('orders', 'allTable', 'area'
13 ));}
14

```

:Edit Order Function *

```

1 // عرض بيانات الطلب في شاشة الخاصة بالتعديل على بيانات الطلب
2 public function edit($id)
3 {
4     // عرض جميع اسماء اقسام الطاولات //
5     $area = area_sections::all();
6     // عرض جميع ارقام الطاولات //
7     $allTable = table::all();
8     // عرض جميع اسماء اقسام قائمة الطعام //
9     $menu = menu_section::all();
10    // عرض جميع اسماء عناصر قائمة الطعام //
11    $items = menu_item::all();
12    // عرض بيانات الطلب من خلال رقم الطلب //
13    $orders = order::find($id);
14    // عرض بيانات عناصر الطلب من خلال رقم الطلب //
15    $order_items = order_item::where('order_id', '=', $id)->get();
16    // الذهاب الى شاشة تعديل بيانات الطلب مع تنفيذ عرض بيانات الطلب من خلال رقم الطلب //
17    return view('/order-update', compact('orders', 'menu', 'items', 'allTable', 'area', 'order_items'
18 )); }

```

Update Order View *

```

1  <!-- route /order/update تضليل كود تتعديل الطلب -->
2  <form action="/order/update" method="post">
3      {{csrf_field()}}
4      {{method_field('PUT')}}
5      <!-- start order form-->
6      <input type="text" name="order_id" value="{{($orders->id)}}>
7      <div class="order-form">
8
9
10     <select list="area-place" name="section_name" id="area_number" placeholder="area section *" onchange="">
11         <option value="{{($orders->section_name)}}" selected>{{ $orders->section->section_name }}</option>
12         <!-- عرض اسم القسم المختار -->
13         <!-- عملية ظهور اسماء القسم وجود اكثرب من قسم يتم استخدام كمبليت تكرار -->
14         @foreach($area as $r)
15             <option value="{{ $r->id }}> {{ $r->section_name }}</option> <!-- عرض اسم القسم -->
16         @endforeach
17     </select>
18
19     <select list="table-id" name="table_id" id="clear_table" placeholder="table number "*>
20         <option value="{{($orders->table_id)}}" selected>{{ $orders->number->table_number }}</option>
21         <!-- عرض رقم الطاولة -->
22         <!-- عملية ظهور ارقام الطاولات وجود اكثرب من طاولة يتم استخدام كمبليت تكرار -->
23         @foreach($allTable as $item)
24             <!-- لعرض رقم الطاولة -->
25             <option value="{{ $item->id }}> {{ $item->area_id }} hide_table> {{ $item->table_number }}</option>
26         @endforeach
27     </select>
28
29     <select list="menu-section" id="item_name" placeholder="menu section *" onchange="">
30         <option value="" disabled selected>select menu</option>
31         <!-- عرض اسم القسم وجود اكثرب من قسم يتم استخدام كمبليت تكرار -->
32         @foreach($menu as $m)
33             <!-- لعرض اسم القسم -->
34             <option value="{{ $m->id }}> {{ $m->section_name }}</option>
35         @endforeach
36     </select>
37
38     <!-- عملية استدعاء سعر العنصر وجود اكثرب من سعر عنصر يتم استخدام كمبليت تكرار -->
39     @foreach($items as $item)
40         <!-- عرض سعر العنصر -->
41         <input class="{{ $item->item_name }}" type="hidden" value="{{ $item->price }}>
42     @endforeach
43
44     <!-- عملية استدعا، رقم العنصر وجود اكثرب من رقم عنصر يتم استخدام كمبليت تكرار -->
45     @foreach($items as $item)
46         <!-- عرض رقم العنصر -->
47         <input class="id_{{ $item->item_name }}" type="hidden" value="{{ $item->id }}>
48     @endforeach
49
50
51     <select list="item-id" id="clear_item" placeholder="menu item *>
52         <option value="" disabled selected>select menu item</option>
53         <!-- عملية ظهور اسم العنصر الخاص بالقسم وجود اكثرب من عنصر يتم استخدام كمبليت تكرار -->
54         @foreach($items as $item)
55             <!-- لعرض اسم العنصر -->
56             <option value="{{ $item->item_name }}> {{ $item->section_id }} hide_table option_item> {{ $item->item_name }}</option>
57         @endforeach
58     </select>
59
60     <input type="date" name="order_date" readonly value="{{($orders->order_date)}}> class="right-side-1" />
61     <input type="time" name="order_time" readonly value="{{($orders->order_time)}}> class="right-side-2" />
62
63     <input type="text" value="{{($orders->notes)}}> name="notes" placeholder="any notes" class="right-side-4" />
64     <input type="text" value="{{($orders->sub_total)}}> name="sub_total" class="sub_total right-side-3" placeholder="sub-total" class="right-side-3" readonly />
65
66     <button type="submit">update</button>
67 </div>
68 <!-- end order form-->
69
70 <!-- start items form-->
71 <div class="items-form">
72     <table>
73         <thead>
74             <tr>
75                 <th>item name</th>
76                 <th>price</th>
77                 <th>quantity</th>
78                 <th>total</th>
79                 <th>ignore</th>
80             </tr>
81         </thead>
82         <tbody>
83             <!-- يتم هنا اضافة العنصر جديد في حالة التعديل -->
84             <!-- استدعاء العنصر العصابة التي اطلب -->
85             <?php
86                 $num = 0;
87             ?>
88             @foreach($order_items as $order)
89             <?php $num++; ?>
90             <tr class="new-tr">
91                 <td> {{($order->item->item_name)}<input type="hidden" name="item_id[]" value="{{($order->item_id)}}><input type="hidden" name="id[]" value="{{($order->id)}}></td>
92                 <td> {{($order->item->price)}<input type="hidden" name="price[]" value="{{($order->item->price)}}></td>
93                 <td>{{input class="quantity" type="number" name="quantity[] value="{{($order->quantity)}} data-price="{{($order->item->price)}} data-total=<?php echo $num; ?>" min="1"></td>
94                 <td>{{input class="total num<?php echo $num; ?>" name="total[]" type="number" readonly value="{{($order->total)}}}</td>
95
96                 <td>
97                     <a class="btn btn-danger delRow text-white " data-num=<?php echo $num; ?>><i class="fas fa-trash"></i><span class="name_item_hide">{{($order->item->item_name)}}</span></a>
98                     <input type="hidden" name="state[]" value="unDelete">
99                 </td>
100             </tr>
101
102             @endforeach
103         </tbody>
104     </table>
105   </div>
106 </form>
107 <!-- end items form-->
```


:Update Function *

```

1 // عملية التحقق من البيانات المدخلة وتنفيذ تعديل الطلب
2 public function update(Request $request)
3 {
4     // التتحقق من البيانات //
5     $validator = Validator::make($request->all(), [
6         'section_name' => 'required',
7         'table_id'      => 'required',
8     ], [
9         'section_name.required'  => 'Please Enter Section Name <br>',
10        'table_id.required'      => 'Please Enter Table number <br>',
11    ]);
12
13     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات
14     if ($validator->fails()) {
15         return back()->with('toast_error', $validator->messages()->all())->withInput();
16     }
17
18     // استدعاء رقم الطاولة من جدول table وذلك لربطها مع رقم الطاولة في جدول الطلب //
19     $t_id = table::where('id', $request->table_id)->first()->id;
20
21     // الكود الخاص بتنفيذ تعديل بيانات الطلب من خلال التأكيد من رقم الطلب //
22     $data_order = order::where('id', $request->order_id);
23     $data_order->update([
24         'order_date' => $request->order_date,
25         'order_time' => $request->order_time,
26         'section_name' => $request->section_name,
27         'notes' => $request->notes,
28         'sub_total' => $request->sub_total,
29         'table_id' => $t_id,
30     ]);
31
32
33     // الكود الخاص بتنفيذ تعديل عناصر الطلب الى جدول //
34     for ($i = 0; $i < count($request->item_id); $i++) {
35
36         $state = $request->state[$i];
37         if ($state == 'unDelete') {
38
39             $item_data = order_item::where('id', $request->id[$i]);
40             $item_data->update([
41                 'quantity' => $request->quantity[$i],
42                 'total' => $request->total[$i],
43             ]);
44         }
45
46
47         // الكود الخاص بتنفيذ اضافة عناصر على عناصر الطلب الاساسي الى جدول //
48         else if ($state == 'newRow') {
49
50             order_item::create([
51                 'order_id' => $request->order_id,
52                 'item_id' => $request->item_id[$i],
53                 'quantity' => $request->quantity[$i],
54                 'total' => $request->total[$i],
55             ]);
56         } else {
57
58             // الكود الخاص بتنفيذ حذف عنصر الطلب من جدول //
59             order_item::where('id', $request->id[$i])->delete();
60         }
61     }
62 }
63
64 // الذهاب الى شاشة قائمة الطلبات مع تنفيذ تعديل الطلب مع ظهور رسالة (تم تعديل الطلب) //
65 // return redirect('order')->with('toast_success', 'Order updated!');
66

```

الذهاب الى شاشة قائمة الطلبات مع تنفيذ تعديل الطلب مع ظهور رسالة (تم تعديل الطلب) // return redirect('order')->with('toast_success', 'Order updated!');

:Delete View *

```

1  <!-- تنفيذ كود الحذف الخاص بالطلب -->
2      <form action="order/destroy" method="post">
3          {{ method_field('delete') }}
4          {{ csrf_field() }}

5          <div class="form-group">
6              <p>Do you sure delete Order</p>
7              <label for="exampleInputEmail1">Table Number</label>
8              <input type="hidden" name="id" id="id" value="">
9              <input type="text" class="form-control" id="table_number" name="table_number" autocomplete="off">
10             </div>
11         </div>
12
13         <div class="modal-footer">
14             <button type="submit" class="btn btn-success">Delete</button>
15             <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16         </div>
17     </form>

```

```

1  <script>
2      // الكود الخاص بإظهار رقم الطاولة في شاشة الحذف
3      $('#modaldemo9').on('show.bs.modal', function(event) {
4          var button = $(event.relatedTarget)
5          var id = button.data('id')
6          var table_number = button.data('table_number')
7          var modal = $(this)
8          modal.find('.modal-body #id').val(id);
9          modal.find('.modal-body #table_number').val(table_number
10      });
11  </script>

```

:Delete Function *

```

1  // الكود الخاص بعملية الحذف على بيانات الطلب
2  public function destroy(Request $request)
3  {
4      $id = $request->id;
5      order::find($id)->delete();
6      return redirect('order')->with('toast_success', 'Order deleted!');
7  };;
}
```

:link to Invoice Screen *

-- الينك الخام بإرسال بيانات الطلب في شاشة الفاتورة --!

```
1 <td><a class="checkout" href="{{route('bill.edit',$order['id'])}}">{{$order->tnumber->table_number}}</a></td>
```

:Edit Invoice Function *

```
1 public function edit($id)
2 {
3     $orders = order::find($id);
4     $order_items = order_item::where('order_id', '=', $id)->get
5     ();
6     return view('bill', compact('orders', 'order_items'));
}
```

:Save Invoice View *

```

1 <!-- route bill.store --> تنفيذ كود اخفاقة فاتورة
2 <form action="{{route('bill.store')}}" method="post">
3 {{csrf_field()}}
4
5 <button type="submit">save</button>
6 <!-- تنفيذ كود طباعة فاتورة -->
7 <button class="print" id="print_Button" onclick="printDiv()">print</button>
8 <!-- start inside order form-->
9 <div class="inside-order-form" id="print">
10
11 <h1>checkout</h1>
12
13 <div class="logo"> </div>
14
15 <input type="hidden" value="{{($orders->id)}}" name="order_id">
16 <h4 class="bill-top-1">table number <span><input type="number"
17 name="table_number" value="{{($orders->table_number)}}> class="form-input" readonly style="width: 100px;"></span></h4>
18 <h4 class="bill-top-2">date<span><input name="check_date" value="{{date('Y-m-d')}}> readonly class="form-input" style="width: 150px;"></span></h4>
19 <h4 class="bill-top-3">time<span><input name="check_time" value="{{date('H:i:s')}}> readonly class="form-input" style="width: 150px;"></span></h4>
20 <!-- hr class="line-1-bill" -->
21 <!--start items form-->
22 <div class="items-form">
23 <table>
24 <thead>
25 <tr>
26 <th>item name</th>
27 <th>price</th>
28 <th>quantity</th>
29 <th>total</th>
30 </tr>
31 </thead>
32 <!-- خرون بيانات الطلب -->
33 <tbody>
34 @foreach($order_items as $order)
35 <tr>
36 <td>$order->item->item_name</td>
37 <td>$order->item->price</td>
38 <td>$order->quantity</td>
39 <td>$order->total</td>
40 </tr>
41 @endforeach
42 </tbody>
43 </table>
44 <hr class="line-2-bill">
45 <h4 class="bill-last-bottom-1">sub total <span><input type="text" id="sub" class="sub_total form-input" readonly value="{{($orders->sub_total)}}> class="form-input"></span></h4><br><br><br>
46 <!-- تنفيذ العمليات الحسابية للوصول الى اجمالي الفاتورة -->
47 @php
48 $services = 12;
49 $tax = 5;
50 $sum1 = $orders->sub_total * $services / 100 ;
51 $sum2 = $orders->sub_total * $tax / 100 ;
52 $sum = $sum1 + $sum2 ;
53 $total = $orders->sub_total + $sum ;
54 @endphp
55 <h4 class="bill-last-bottom-1">services value <span><input type="text" id="services" readonly value="{{$services}}> name="services_val" class="services form-input"><span class="pe">%</span></span></h4><br><br>
56 <h4 class="bill-last-bottom-2">tax value <span><input type="text" id="tax" readonly value="{{$tax}}> name="tax_val" class="tax form-input"><span class="per">%</span></span></h4><br><br><br>
57 <h4 class="bill-last-bottom-3">total amount <span><input type="text" id="total" value="{{$total}}> readonly name="total_amount" class="total form-input"></span></h4><br><br><br>
58 </div>
59 <!--end items form-->
60 </div>
61 </form>

```

:Save Invoice Function *

```

1  public function store(Request $request)
2      {
3          $order_id = $request->order_id;
4          $order_exists = check_out::where('order_id', '=', $order_id)->exists
5      ();
6          if($order_exists){
7              return 'this invoice is already exists';
8          }
9          else{
10
11              check_out::create([
12                  'invoice_date' =>$request->check_date,
13                  'invoice_time' =>$request->check_time,
14                  'service' =>$request->services_val,
15                  'tax' =>$request->tax_val,
16                  'amount_total' =>$request->total_amount,
17                  'order_id' =>$request->order_id,
18                  'table_id' =>$request->table_number,
19                  'user'=>(Auth::user()->name),
20                  ]);
21
22          return redirect('bill');
23      }
24  }

```

:Print Invoice Function *

```

1  <script>
2  كود تنفيذ طباعة الفاتورة //
3  function printDiv() {
4      var printContents = document.getElementById('print').innerHTML
5      ;
6      var originalContents = document.body.innerHTML;
7      document.body.innerHTML = printContents;
8      window.print();
9      document.body.innerHTML = originalContents;
10     location.reload();
11  }
12  </script>

```

:Show Invoice View *

```

1 <!--start bill controls form-->
2   <div class="bill-controls">
3     <div class="search-part">
4       <table>
5         <thead>
6           <tr>
7             <th>ID</th>
8             <th>order number</th>
9             <th>table number</th>
10            <th>invoice date</th>
11            <th>invoice time</th>
12            <th>total amount</th>
13            <th>user name</th>
14            <th>Delete</th>
15          </tr>
16        </thead>
17        <tbody>
18        <!-- عرض بيانات الفاتورة -->
19        @foreach($check as $c)
20          <tr>
21            <td>{{$c->id}}</td>
22            <td>{{$c->order_id}}</td>
23            <td>{{$c->table_id}}</td>
24            <td>{{$c->invoice_date}}</td>
25            <td>{{$c->invoice_time}}</td>
26            <td>{{$c->amount_total}}</td>
27            <td>{{$c->user}}</td>
28            <!-- لينك حذف الفاتورة -->
29            <td><a class="del" href="#modaldemo9" data-toggle="modal" data-id="{{$c->id}}" data-table_number="{{$c->table_i
30 d }}><i class="fas fa-trash"></i></a>
31          </td>
32        </tr>
33      @endforeach
34    </tbody>
35  </table>
36 </div>

```

:Show Invoice Function *

```

1 public function index()
2 {
3     $check = check_out::all();
4     return view('bill-controls', compact('check',$check
5 ));}
6

```

:Delete Invoice View *

```

1  <!-- bill/destroy -->
2      <form action="bill/destroy" method="post">
3          {{ method_field('delete') }}
4          {{ csrf_field() }}

5
6          <div class="form-group">
7              <p>Do you sure delete invoice</p>
8              <label for="exampleInputEmail1">Table Number</label>
9              <input type="hidden" name="id" id="id" value="">
10             <input type="text" class="form-control" id="table_number" name="table_number" autocomplete="off"
11         </div>
12
13         <div class="modal-footer">
14             <button type="submit" class="btn btn-success">Delete</button>
15             <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16         </div>
17     </form>

```

```

1  <script>
2 // الكود الخامس باظهار رقم الطاولة في شاشة الحذف
3 $('#modaldemo9').on('show.bs.modal', function(event) {
4     var button = $(event.relatedTarget)
5     var id = button.data('id')
6     var table_number = button.data('table_number')
7     var modal = $(this)
8     modal.find('.modal-body #id').val(id);
9     modal.find('.modal-body #table_number').val(table_number
10 );  })
11
12 </script>
13

```

:Delete Invoice Function *



```
1 public function destroy(Request $request
2 ) {
3     $id = $request->id;
4     check_out::find($id)->delete();
5     return redirect('bill');
6 }
```

:Save Employees View *

```
1 public function create() // تشغيل صفحة الموظف
2 {
3     return view('employees');
4 }
```



```
1 <!-- route employees.store --> تفاصيل كود اضافة الموظفين
2 <form action="{{route('employees.store')}}" method="post" autocomplete="off">
3 {{csrf_field()}}
4     <input type="text" name="emp_name" placeholder="name *"/>
5     <input type="text" name="emp_phone" placeholder="phone *"/>
6     <input type="text" name="emp_address" placeholder="address *"/>
7     <input type="data" readonly value="{{date('Y-m-d')}}" name="emp_datetim" placeholder="hiredate *"/>
8     <input type="number" name="emp_salary" placeholder="salary *"/>
9
10    <select list="job-type" id="job-type" name="emp_type" placeholder="job type *">
11        <option disabled selected>select job type</option>
12        <option value="manager">manager</option>
13        <option value="cashier">cashier</option>
14        <option value="waiter">waiter</option>
15    </select>
16
17    <select list="emp-gender" name="emp_gender" placeholder="gender *">
18        <option disabled selected>select gender</option>
19        <option value="male">male</option>
20        <option value="female">female</option>
21    </select>
22
23    <button type="submit" value="save">save</button>
24    <button type="reset" value="clear" class="btn-clear">clear</button>
25 </form>
```

:Save Employees Function *

```

1 // عملية التحقق من البيانات وعملية اضافة الموظفين //
2 public function store(Request $request)
3 {
4
5     // التتحقق من البيانات //
6     $validator = Validator::make($request->all(), [
7         'emp_name'      => 'required|min:8',
8         'emp_phone'     => 'required|numeric|digits:11|starts_with:0',
9         'emp_address'   => 'required|min:8',
10        'emp_salary'    => 'required|integer|min:1500',
11        'emp_type'      => 'required',
12        'emp_gender'    => 'required',
13
14    ], [
15        // -----
16        'emp_name.required'    =>'Please Enter Employee Name <br>',
17        'emp_name.min'          =>'Error Entering Name This Name is Less Than 8 Letters <br>',
18        //-----
19
20        //-----
21        'emp_phone.required'   =>'Please Enter Employee Phone <br>',
22        'emp_phone.numeric'    =>'Error Entering Number Phone Please Enter Number Phone <br>',
23        'emp_phone.digits'     =>'Error Entering Number Phone This Number Phone is Less Than Or Greater Than 11 Number <br>'
24        'emp_phone.starts_with'=>'Error Entering Number Phone Please Enter Number Phone <br>',
25        //-----
26
27        //-----
28        'emp_address.required'=>'Please Enter Employee Address <br>',
29        'emp_address.min'       =>'Error Entering Address This Address is Less Than 8 Letters <br>',
30        //-----
31
32        //-----
33        'emp_salary.required'  =>'Please Enter Salary <br>',
34        'emp_salary.integer'   =>'Error Entering Salary Please Enter Salary <br>',
35        'emp_salary.min'        =>'Error Entering Salary This Salary is Less Than 1500 <br>',
36        //-----
37
38
39        'emp_type.required'    =>'Please Enter Job Type <br>',
40        'emp_gender.required'  =>'Please Enter Gender <br>',
41    ]);
42
43     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
44     if ($validator->fails()) {
45         return back()->with('toast_error', $validator->messages()->all())->withInput();
46     }
47
48     // الكود الخاص بعملية اضافة الموظفين //
49     $emp = new employee();
50     $emp->name=$request->input('emp_name');
51     $emp->phone=$request->input('emp_phone');
52     $emp->address=$request->input('emp_address');
53     $emp->hire_date=$request->input('emp_datetime');
54     $emp->salary=$request->input('emp_salary');
55     $emp->job_type=$request->emp_type;
56     $emp->gender=$request->emp_gender;
57     $emp->save();
58     return redirect('employees')->with('toast_success', 'Add Employee Created!');
59 }

```

Show Employees View *

```

1 <div class="table table-hover">
2     <table>
3         <thead>
4             <tr>
5                 <th>ID</th>
6                 <th>employee name</th>
7                 <th>phone</th>
8                 <th>address</th>
9                 <th>hire-date</th>
10                <th>salary</th>
11                <th>job-type</th>
12                <th>gender</th>
13                <th>update</th>
14                <th>delete</th>
15            </tr>
16        </thead>
17        <tbody>
18            <!-- استرجاع جميع بيانات الموظفين -->
19            <?php $i=0; ?>
20            @foreach($emp as $m)
21            <?php $i++; ?>
22            <tr>
23                <td>{{$i}}</td>
24                <td>{{$m->name}}</td>
25                <td>{{$m->phone}}</td>
26                <td>{{$m->address}}</td>
27                <td>{{$m->hire_date}}</td>
28                <td>{{$m->salary}}</td>
29                <td>{{$m->job_type}}</td>
30                <td>{{$m->gender}}</td>
31                <td>
32                    <!-- اللينك الخاص بإرسال بيانات الموظف في شاشة التعديل -->
33                    <a href="{{route('employees.edit', $m['id'])}}><i class="fas fa-edit"></i></a>
34                </td>
35                <td>
36                    <!-- اللينك شاشة المسح -->
37                    <a href="#modaldemo9" data-toggle="modal" data-id="{{$m->id}}" data-emp_name="{{$m->name}}"
38 e }" data-toggle="modal" class="del" href="#modaldemo9"><i class="fas fa-trash"></i></a>
39                </td>
40            </tr>
41        @endforeach
42        </tbody>
43    </table>
44</div>

```

:Show Employees Function *

```
● ● ●
1 public function index() // اول عملية يتم تنفيذها
2 {
3     // استرجاع جميع بيانات الموظفين //
4     $emp = employee::all();
5     return view('employees-controls',compact('emp'))
6 }) ; }
```

:Search Employees View *

```
● ● ●
1 <form method="get" action="/search_emp">
2     {{csrf_field()}}
3     <input class="txt_search" autocomplete="off" name="search" type="search" placeholder="search for employees" />
4     <button type="submit" class="btn_search">search</button>
5 </form>
```

:Search Employees Function *

```
● ● ●
1 public function search_emp(Request $request)
2 {
3     $search = $request->get('search');
4     $emp = employee::select('*')->where('name',$search)
5     ->orwhere('phone',$search)
6     ->orwhere('job_type',$search)->get();
7     return view('employees-controls',compact('emp'));
8 }
9 }
```

:Edit Employees Function *



```

1  استرجاع بيانات الموظف في شاشة الخاصة بالتعديل على بيانات الموظف // ف
2  //public function edit($id)
3  {
4      $emps = employee::find($id);
5      return view('/employees-update',compact('emps','id'));
6  }

```

:Update Employees View *



```

-- تنفيذ الكود من خلال الرقم الموظف الجزء الخامس بعملية التعديل
1  <!-- action('EmployeeController@update',$id) -->
2  <form method="post" action="{{action('EmployeeController@update',$id)}}" autocomplete="off">
3      {{method_field('patch')}}
4      {{csrf_field()}}
5      <input type="hidden" value="{{$emps->id}}" name="id" placeholder="ID *"/>
6      <input type="text" value="{{$emps->name}}" name="emp_name" placeholder="name *"/>
7      <input type="text" value="{{$emps->phone}}" name="emp_phone" placeholder="phone *"/>
8      <input type="text" value="{{$emps->address}}" name="emp_address" placeholder="address *"/>
9      <input type="date" value="{{$emps->hire_date}}" name="emp_datetime" placeholder="hiredate *"/>
10     <input type="text" value="{{$emps->salary}}" name="emp_salary" placeholder="salary *"/>
11
12     <select list="job-type" id="job-type" name="emp_type" placeholder="job type *">
13         <option value="{{$emps->job_type}}" selected>{{$emps->job_type}}</option>
14         <option value="manager">manager</option>
15         <option value="cashier">cashier</option>
16         <option value="waiter">waiter</option>
17     </select>
18
19     <select list="emp-gender" name="emp_gender" placeholder="gender *">
20         <option value="{{$emps->gender}}" selected>{{$emps->gender}}</option>
21         <option value="male">male</option>
22         <option value="female">female</option>
23     </select>
24
25     <!-- <p class="text-white bg-danger h4 mt-3 w-50 pl-2 ">test</p> -->
26     <button type="submit">update</button>
27     <button type="reset">clear</button>
28 </form>

```

:Update Employees Function *

```

1 // عملية التحقق من البيانات وعملية التعديل على بيانات الموظف //
2 public function update(Request $request)
3 {
4
5     // التتحقق من البيانات //
6     $validator = Validator::make($request->all(), [
7         'emp_name'      => 'required|min:8',
8         'emp_phone'     => 'required|numeric|digits:11|starts_with:0',
9         'emp_address'   => 'required|min:8',
10        'emp_salary'    => 'required|integer|min:1500',
11        'emp_type'      => 'required',
12        'emp_gender'    => 'required',
13
14    ], [
15        // -----
16        'emp_name.required'      =>'Please Enter Employee Name <br>',
17        'emp_name.min'           =>'Error Entering Name This Name is Less Than 8 Letters <br>',
18        //-----
19
20        //-----
21        'emp_phone.required'     =>'Please Enter Employee Phone <br>',
22        'emp_phone.numeric'      =>'Error Entering Please Enter Number <br>',
23        'emp_phone.digits'       =>'Error Entering Number This Number is Less Than Or Greater Than 11 Number <br>',
24        'emp_phone.starts_with'  =>'Error Entering Number Phone Please Enter Number Phone <br>',
25        //-----
26
27        //-----
28        'emp_address.required'   =>'Please Enter Employee Address <br>',
29        'emp_address.min'        =>'Error Entering Address This Address is Less Than 8 Letters <br>',
30        //-----
31
32        //-----
33        'emp_salary.required'   =>'Please Enter Salary <br>',
34        'emp_salary.integer'    =>'Error Entering Salary Please Enter Salary <br>',
35        'emp_salary.min'         =>'Error Entering Salary This Salary is Less Than 1500 <br>',
36        //-----
37
38        'emp_type.required'     =>'Please Enter Job Type <br>',
39        'emp_gender.required'   =>'Please Enter Gender <br>',
40
41    ]);
42
43
44     // اظهار رسالة الخطأ لو كان هناك ادخال خطأ في البيانات //
45     if ($validator->fails()) {
46         return back()->with('toast_error', $validator->messages()->all())->withInput();
47     }
48
49     // الكود الخام بعملية التعديل على بيانات الموظفين //
50     $id = $request->input('id');
51     $edit = employee::where('id', '=', $id);
52     $edit->update([
53         'name'=>$request->input('emp_name'),
54         'phone'=>$request->input('emp_phone'),
55         'address'=>$request->input('emp_address'),
56         'hire_date'=>$request->input('emp_datetime'),
57         'salary'=>$request->input('emp_salary'),
58         'job_type'=>$request->input('emp_type'),
59         'gender'=>$request->input('emp_gender')
60     ]);
61     return redirect('employees')->with('toast_success', 'Update successfully!');
62
63 }

```

:Delete Employees View *

```

1 <!-- employees/destroy -->
2     <form action="employees/destroy" method="post">
3         {{ method_field('delete') }}
4         {{ csrf_field() }}

5
6         <div class="form-group">
7             <p>Do you sure delete employee </p>
8             <label for="exampleInputEmail1">Employee Name</label>
9             <input type="hidden" name="id" id="id" value="">
10            <input type="text" class="form-control" id="emp_name" name="emp_name" autocomplete="off">
11        </div>
12
13        <div class="modal-footer">
14            <button type="submit" class="btn btn-success">Delete</button>
15            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
16        </div>
17    </form>

```

```

1 <script>
2 // الكود الخاص بإظهار الاسم الموظف في شاشة المنسج
3 $('#modaldemo9').on('show.bs.modal', function(event
4 ) {
5     var button = $(event.relatedTarget)
6     var id = button.data('id')
7     var emp_name = button.data('emp_name')
8     var modal = $(this)
9     modal.find('.modal-body #id').val(id);
10    modal.find('.modal-body #emp_name').val(emp_name
11 );
12 })
13 </script>

```

:Delete Employees Function *



```
1 // الكود الخامس بعملية المسح على بيانات الموظفين //
2 public function destroy(Request $request)
3 {
4     $id = employee::findOrFail($request->id);
5     $id->delete();
6     return redirect('employees')->with('toast_success', 'Delete successfully!');
7 };
```

:Search Report Employees View *

```

1  <!-- <a href="{{url('export_reserve')}}" class="export">export excel</a> -->
2  <form method="post" action="/search_employee" role="search">
3      {{csrf_field()}}
4      <input class="txt_search" autocomplete="off" name="search_report" type="search" placeholder="search for employees" /
5 >      <input class="txt_search1" name="search_start" type="date" value="{{ $search_start ?? 'Select Date' }}" />
6      <input class="txt_search2" name="search_end" type="date" value="{{ $search_end ?? 'Select Date' }}" />
7
8      <select list="job_type" name="type" placeholder="Job Type">
9          <option value="{{ $type ?? 'Select All' }}" disabled selected>{{ $type ?? 'Select All' }}</option>
10         <option value="All Employee">All Employee</option>
11     </select>
12
13     <button type="submit" class="btn_search">search</button>
14 </form>

```

:Search Report Employees Function *

```

1  public function search_emp(Request $request)
2  {
3
4      if($request->search_report){
5
6          $search = $request->search_report;
7          $employee = employee::select('*')->where('name', '=', $search)
8              ->orwhere('phone', '=', $search )
9              ->orwhere('job_type', '=', $search )->get();
10         return view('reports.emp_report')->withDetails($employee);
11     }
12     else if($request->type){
13
14         $type = $request->type;
15         $employee = employee::select('*')->get();
16         return view('reports.emp_report', compact('type'))->withDetails($employee);
17     }
18
19     else if($request->search_start){
20
21         $search_start = date($request->search_start);
22         $employee = employee::select('*')->where('hire_date', '=', $request->search_start)->get();
23         return view('reports.emp_report', compact('search_start'))->withDetails($employee);
24     }
25
26     else{
27
28         $search_start = date($request->search_start);
29         $search_end = date($request->search_end);
30         $employee = employee::whereBetween('hire_date', [$search_start, $search_end])->get();
31         return view('employees-controls', compact('search_start', 'search_end'))->withDetails($employee
32     );
33 }

```

:Show Report Employees View *

```

1  <div class="search-part">
2      <table class="table" id="table"
3          data-show-export="true"
4          data-toolbar="#toolbar"
5          data-show-columns="true"
6      >
7      <thead>
8          <tr>
9              <th>ID</th>
10             <th>employee name</th>
11             <th>phone</th>
12             <th>address</th>
13             <th>hire-date</th>
14             <th>salary</th>
15             <th>job-type</th>
16             <th>gender</th>
17         </tr>
18     </thead>
19     <tbody>
20     @if(isset($details))
21         <?php $i=0; ?>
22         @foreach($details as $employee)
23             <?php $i++; ?>
24             <tr>
25                 <td>{{$i}}</td>
26                 <td>{{$employee->name}}</td>
27                 <td>{{$employee->phone}}</td>
28                 <td>{{$employee->address}}</td>
29                 <td>{{$employee->hire_date}}</td>
30             >
31                 <td>{{$employee->salary}}</td>
32                 <td>{{$employee->job_type}}</td>
33                 <td>{{$employee->gender}}</td>
34             </tr>
35         @endforeach
36     @endif
37     </tbody>
38 </table>

```

:Show Report Employees Function *



```
1 public function index()
2 {
3     return view('reports.emp_report'
4 ); }
```

:Search Report Reservation View *

```

1  <!-- <a href="{{url('export_reserve')}}" class="export">export excel</a> -->
2  <form method="post" action="/search_reservation" role="search">
3      {{csrf_field()}}
4
5      <label class="customer_date">searsh for customer date</label>
6      <input class="txt_search" autocomplete="off" name="search_date" type="date" />
7
8      <label class="customer_date1">searsh for start date</label>
9      <input type="date" name="search_start" value="" class="invoice_date_start" />
10
11     <label class="customer_date2">searsh for end date</label>
12     <input type="date" name="search_end" value="" class="invoice_date_end" />
13
14     <select class="job" list="job_type" name="type" placeholder="Job Type">
15         <option value="{{ $type ?? 'Select All' }}" disabled selected>{{ $type ?? 'Select All' }}</option>
16         <option value="All Employee">All Reservations</option>
17     </select>
18     <button type="submit" class="btn_search">search</button>
19 </form>

```

:Search Report Reservation Function *

```

1  public function search_res(Request $request)
2  {
3
4      if($request->type){
5
6          $type = $request->type;
7          $reservation = reservation::select('*')->get();
8          return view('reports.res_report',compact('type'))->withDetails($reservation );
9      }
10
11     else if($request->search_date){
12
13         $search_date = $request->search_date;
14         $reservation = reservation::select('*')->where('customer_date' , '=' , $request->search_date)->get
15     ());
16         return view('reports.res_report',compact('search_date'))->withDetails($reservation );
17     }
18
19     else if($request->search_start){
20
21         $search_start = date($request->search_start);
22         $reservation = reservation::select('*')->where('reserve_date' , '=' , $request->search_start)->get
23     ());
24         return view('reports.res_report',compact('search_start'))->withDetails($reservation );
25     }
26
27     else{
28
29         $search_start = date($request->search_start);
30         $search_end = date($request->search_end);
31         $reservation = reservation::whereBetween('reserve_date' , [$search_start,$search_end])->get();
32         return view('reports.res_report',compact('search_start','search_end'))->withDetails($reservation );
33     }
34 }

```

:Show Report Reservation View *

```

1  <div class="search-part">
2      <table class="table" id="table"
3          data-show-export="true"
4          data-toolbar="#toolbar"
5          data-show-columns="true"
6      >
7      <thead>
8          <tr>
9              <th>ID</th>
10             <th>customer name</th>
11             <th>customer phone</th>
12             <th>count of person</th>
13             <th>reserved date</th>
14             <th>customer date</th>
15             <th>customer time</th>
16             <th>area place</th>
17             <th>table number</th>
18             <th>user name</th>
19         </tr>
20     </thead>
21     <tbody>
22     @if(isset($details))
23         <?php $i=0; ?>
24         @foreach($details as $reservation)
25             <?php $i++; ?>
26             <tr>
27                 <td>{{$i}}</td>
28                 <td>{{$reservation->cust_name}}</td>
29                 <td>{{$reservation->cust_phone}}</td>
30                 <td>{{$reservation->count_person}}</td>
31                 <td>{{$reservation->reserve_date}}</td>
32                 <td>{{$reservation->customer_date}}</td>
33                 <td>{{$reservation->customer_time}}</td>
34                 <td>{{$reservation->section->section_name}}</td>
35             >
36             <td>{{$reservation->tnumber->table_number}}</td>
37             <td>{{$reservation->user}}</td>
38         </tr>
39     @endforeach
40     @endif
41     </tbody>
42   </table>
43 </div>

```

:Show Report Reservation Function *



```
1 public function index()
2 {
3     return view('reports.emp_report'
4 ); }
```

:Search Report Order View *

```

1 <form method="post" action="/search_order" role="search">
2     {{csrf_field()}}
3     <input class="txt_search" autocomplete="off" name="search_report" type="search" placeholder="Table Number" /
4 >
5     <input type="date" name="search_start" value="" class="invoice_date_start" />
6     <input type="date" name="search_end" value="" class="invoice_date_end" />
7
8     <select list="job_type" name="type" placeholder="Job Type">
9         <option value="{{ $type ?? 'Select All' }}" disabled selected>{{ $type ?? 'Select All' }}</option>
10        <option value="All Orders">All Orders</option>
11    </select>
12    <button type="submit" class="btn_search">search</button>
13 </form>

```

:Search Report Order Function *

```

1 public function search_order(Request $request)
2 {
3     if($request->search_report){
4
5         $search = table::where('table_number', $request->search_report )->first()->id;
6
7         $order = order::select('*')->where('table_id','=',$search)->get();
8         return view('reports.report_order')->withDetails($order);
9     }
10
11     else if($request->type){
12
13         $type = $request->type;
14         $order = order::select('*')->get();
15         return view('reports.report_order',compact('type'))->withDetails($order );
16     }
17
18     else if($request->search_start){
19
20         $search_start = date($request->search_start);
21         $order = order::select('*')->where('order_date' , '=' , $request->search_start)->get();
22         return view('reports.report_order',compact('search_start'))->withDetails($order );
23     }
24
25     else{
26
27         $search_start = date($request->search_start);
28         $search_end = date($request->search_end);
29         $order = order::whereBetween('order_date' , [$search_start,$search_end])->get();
30         return view('reports.report_order',compact('search_start','search_end'))->withDetails($order
31 );      }
32
33 }

```

:Show Report Order View *

```

1 <div class="search-part">
2     <table class="table" id="table"
3         data-show-export="true"
4         data-toolbar="#toolbar"
5         data-show-columns="true"
6     >
7     <thead>
8         <tr>
9             <th>ID</th>
10            <th>section name</th>
11            <th>table number</th>
12            <th>date</th>
13            <th>time</th>
14            <th>sub total</th>
15            <th>user name</th>
16        </tr>
17    </thead>
18    <tbody>
19    @if(isset($details))
20        <?php $i=0; ?>
21        @foreach($details as $order)
22            <?php $i++; ?>
23            <tr>
24                <td>{{$i}}</td>
25                <td>{{$order->section->section_name}}</td>
26            >
27            <td>{{$order->tnumber->table_number}}</td>
28            <td>{{$order->order_date}}</td>
29            <td>{{$order->order_time}}</td>
30            <td>{{$order->sub_total}}</td>
31            <td>{{$order->user}}</td>
32        </tr>
33    @endforeach
34    @endif
35    </tbody>
36    </table>
</div>

```

:Show Report Order Function *



```
1 public function index()
2 {
3     return view('reports.report_order'
4 ); }
```

:Search Report Invoice View *

```

1 <!-- <a href="{{url('export_reserve')}}" class="export">export excel</a> -->
2     <form method="post" action="/search_invoice" role="search">
3         {{csrf_field()}}
4         <input class="txt_search" autocomplete="off" name="search_report" type="search" placeholder="Table Number" /
5     >
6         <input type="date" name="search_start" value="" class="invoice_date_start" />
7         <input type="date" name="search_end" value="" class="invoice_date_end" />
8
9         <select list="job_type" name="type" placeholder="Job Type">
10            <option value="{{ $type ?? 'Select All' }}" disabled selected>{{ $type ?? 'Select All' }}</option>
11            <option value="All Invoice">All Invoice</option>
12        </select>
13        <button type="submit" class="btn_search">search</button>
14    </form>

```

:Search Report Invoice Function *

```

1 public function search_invoice(Request $request)
2 {
3     if($request->search_report){
4
5         $search = table::where('table_number', $request->search_report )->first()->id;
6         $invoice = check_out::select('*')->where('table_id', '=', $search)->get();
7         return view('reports.report_invoice')->withDetails($invoice);
8     }
9
10    else if($request->type){
11
12        $type = $request->type;
13        $invoice = check_out::select('*')->get();
14        return view('reports.report_invoice',compact('type'))->withDetails($invoice );
15    }
16
17    else if($request->search_start){
18
19        $search_start = date($request->search_start);
20        $invoice = check_out::select('*')->where('invoice_date' , '=' , $request->search_start)->get();
21        return view('reports.report_invoice',compact('search_start'))->withDetails($invoice );
22    }
23
24    else{
25
26        $search_start = date($request->search_start);
27        $search_end = date($request->search_end);
28        $invoice = check_out::whereBetween('invoice_date' , [$search_start,$search_end])->get();
29        return view('reports.report_invoice',compact('search_start','search_end'))->withDetails($invoice
30    );}
31
32 }

```

:Show Report Invoice View *

```

1  <div class="search-part">
2      <table class="table" id="table"
3          data-show-export="true"
4          data-toolbar="#toolbar"
5          data-show-columns="true"
6      >
7      <thead>
8          <tr>
9              <th>ID</th>
10             <th>order number</th>
11             <th>table number</th>
12             <th>invoice date</th>
13             <th>invoice time</th>
14             <th>total amount</th>
15             <th>user name</th>
16         </tr>
17     </thead>
18     <tbody>
19     @if(isset($details))
20         <?php $i=0; ?>
21         @foreach($details as $invoice)
22             <?php $i++; ?>
23             <tr>
24                 <td>{{$i}}</td>
25                 <td>{{$invoice->order_id}}</td>
26                 <td>{{$invoice->table_id}}</td>
27                 <td>{{$invoice->invoice_date}}</td>
28             >
29             >
30             >
31                 <td>{{$invoice->invoice_time}}</td>
32                 <td>{{$invoice->amount_total}}</td>
33                 <td>{{$invoice->user}}</td>
34             </tr>
35         @endforeach
36         @endif
37     </tbody>
38     </table>
39 </div>

```

:Show Report Invoice Function *



```
1 public function index()
2 {
3     return view('reports.report_invoice'
4 ); }
```