LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC6001NI – Advance Database Systems Development

Assessment Weightage & Type

40% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Chirag Chandra Gautam

London Met ID: 18029511

College ID: 180017

Assignment Due Date: 19th March 2021

Assignment Submission Date: 19th March 2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Acknowledgement:

We might want to express our extraordinary thanks of appreciation to our module leader and lecturer Mr. Rohit Panday for his lord direction and his guidance to the coursework. He gave us the opportunity to do the assignment and get some knowledge regarding Databases and the problems. We are exceptionally thankful also toward our tutor Mr. Abhisekh Humagain for guiding us through this project and helping us out throughout the project. We are extremely thankful towards all the teachers from this module who allowed us the chance to take a shot at this awesome venture, which helped us to work. It likewise helped us to accomplish learning about numerous new things.

Table of Contents

Table of figures

1. Introduction

This is a report of the coursework which had given the assignment of dissecting, planning and carrying out an online data set application dependent on a given business contextual investigation utilizing Oracle SQL Developer Data Modeler, Oracle SQL Developer and ASP.NET with C# (Visual Studio). According to the contextual investigation, the framework needs to have an appropriate information base for Good Food Hotel which keeps the record of the relative multitude of exchanges.

The given assignment in the coursework were difficult by any means. Acknowledging about the durability of the assignment, this coursework was begun and finished with extraordinary assurance. Since, the fundamental objective of the coursework was to complete every one of the undertakings given in the coursework inside the given timeframe. Another, primary point of the task was to finish every one of the standards of the coursework in the most ideal manner conceivable.

At first, normalization of given figures was done. Then, according to the normalization and case study a proper ER diagram using Oracle SQL Developer Data Modeler was made. Similarly, assumptions according to the ER diagram was also done. Then, from the ER diagram, DDL file and data dictionary was exported to generate the tables and produce a list of attributes for each entity respectively. Likewise, DDL file was imported in Oracle SQL Developer and all the tables were generated. Similarly, all the generated tables were populated with suitable data values using INSERT statement. Finally, implementation of web-based database application was done using ASP.NET with C# which contains different web forms in an attractive dashboard. The conceptual diagram of my report is shown below:

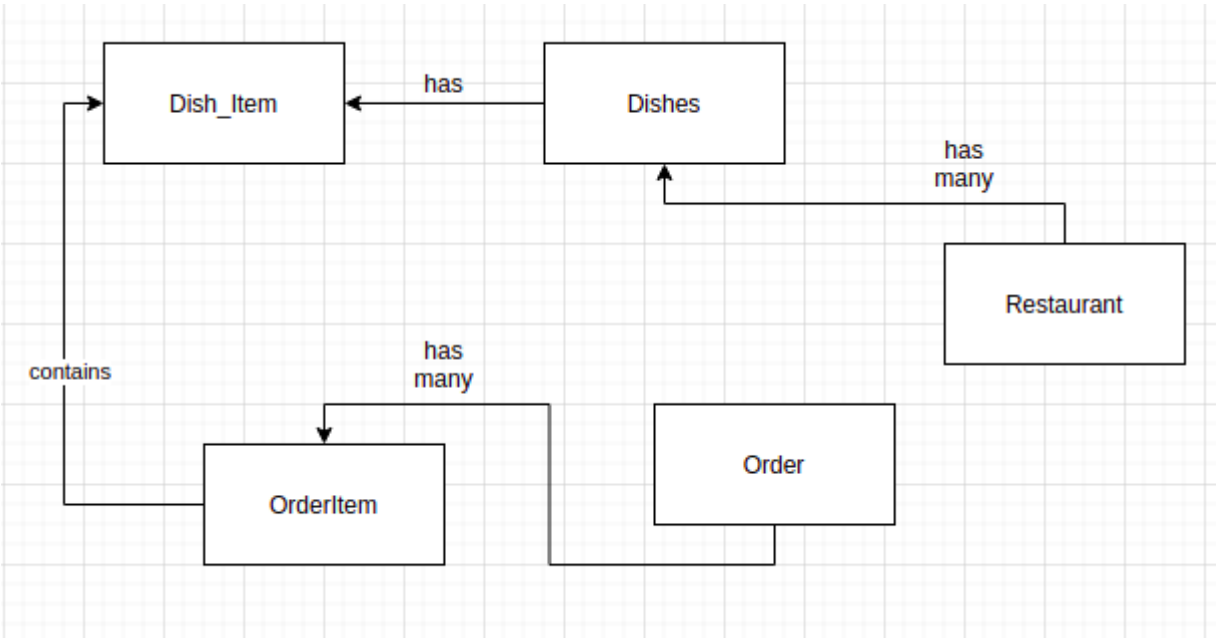


Figure 1: figure of conceptual diagram

The above figure displays the tables in my database structure. The table contains different attributes and entities which is going to be displayed in the Entity Relational Diagram (ERD). The figure is a rough assumption of my database which is going to be displayed in the following reports thoroughly.

2. Normalization:

2.1. Textual Analysis:

i. OrderItem contains many dishes_items which is related using dish_code

ii. One Restaurant can have many dish_item.

iii. One restaurant can have many dishes and many dishes can contain the same dish item.

iv. One Order can have many order items.

v. This relation is set up using Order_OrderItem_.

2.2. Normalization of Figure 1:

1-NF

Rules of 1-NF

- Each cell must have atomic value
- Each record must be unique

Dishes

DISH Code	Dish Name	Local or Another Name	Available on Restaurant
CHKMO	Chicken Momo	Chicken Momo	Royale Luxury
VEGTM	Vegetable Tempura		Royale Luxury
CHKMO	Chicken Momo	Chicken Momo	Taste Buds
MIXCHT	Mixed Chatamari	Nepali Pizza	Local Foodie

Order

SN	Data/Time	Order	Order	Delivery Point	Status	Dish Code	Dish Name	Order Unit	Dish	Line	Restaurant

		Number	Amount						Rate	Total	
2020-09-223334	9th November 2020/14:30	09-000231	2750	Baneshwor	On Time	CHKMO	Chicken Momo	6 Plates	350	2100	Taste Buds
2020-09-223334	9th November 2020/14:30	09-000231	2750	Baneshwor	On Time	MIXCHT	Mixed Chatamari	2 Plate	325	650	Local Foodie
2020-09-223374	9th November 2020/15:20	09-000345	700	Kalanki	Early	CHKMO	Chicken Momo	2 Plates	350	700	Taste Buds
2020-09-223334	9th November 2020/14:40	09-000360	5000	Baluwatar	On Time						
2020-09-223334	9th November 2020/16:50	09-090000	3450	Kalanki	Late						

Tables after 1-NF

Dishes(dish_code,dish_name,local_another_name,available_restaurant)

Order(SN,date_time,order_number,delivery_point,status,dish_code,dish_name,order_unit,dish_rate,line_total,restaurant)

2-NF

Rules of 2-NF

- The table must be in 1-NF
- Single Column Primary key

From the table we can see that we can break the dishes table into dishes and restaurants because the available one does not depend on the primary key dish code.

Restaurant Code	Restaurant Name
RXL51	Royal Luxury
TAB49	Taste Buds
LOC59	Local Foodie

The dishese table can be further splitted inorder to reduce the data redundancy for dish items for restaurants. For example chicken momo repeats for taste buds and royale luxury.

Dish_Item

Dish Code	Dish Name	Local or Another Name
CHKMO	Chicken Momo	Chicken Momo
VEGTM	Vegetable Tempura	
MIXCHT	Mixed Chatamari	Nepali Pizza

Dishes

DISH Code	Restaurant Code
CHKMO	RXL51
VEGTM	RXL51
CHKMO	TAB49
MIXCHT	LOC59

Dish_Item(dish_code(PK),dish_name,alternative_name)
Dishes(dish_code(PK/FK),reasturant_code(PK/FK))
Restaurant(reasturan_code, reasturant_name)

Order(SN,date_time,order_number,delivery_point,status,dish_code,dish_name,order_unit,dish_rate,line_total,restaurant)

3-NF

Rules for 3-NF

- Tables must be in 3NF
- There must not exist any transitive dependency

We can see columns in the order register table. The dish code can be obtained using SN, and the dish code can be used to acquire information about the dish name using the dish code (SN->dish code->dish name). As a result, transitive dependence occurs.
As a result, we divided the table as follows:

For example:

SN	Data/Time	Order Number	Order Amount	Delivery Point	Status	Dish Code	Dish Name	Order Unit	Dish Rate	Line Total	Restaurant
2020-09-223334	9th November 2020/14:30	09-00231	2750	Baneshwor	On Time	CHKMO	Chicken Momo	6 Plates	350	2100	Taste Buds
2020-09-223334	9th November 2020/14:30	09-00231	2750	Baneshwor	On Time	MIXCHT	Mixed Chatamari	2 Plate	325	650	Local Foodie

We can separate the table into Order, OrderItem and OrderRegisterTable

Table Name:Order

SN	Data/Time	Order Number	Order Amount	Delivery Point	Status
2020-09-223334	9th November 2020/14:30	09-00231	2750	Baneshwor	On Time

Table Name: Order Item

Dish Code	Dish Name	Order Unit	Dish Rate	Line Total	Restaurant
CHKMO	Chicken Momo	6 Plates	350	2100	Taste Buds
MIXCHT	Mixed Chatamari	2 Plate	325	650	Local Foodie

Table Name: OrderRegisterTable

SN	DishCode
2020-09-223334	CHKMO
2020-09-223334	MIXCHT

Tables after 3-NF

Dish_Item(dish_code(PK),dish_name,alternative_name)
Dishes(dish_code(PK/FK),reasturant_code(PK/FK))
Restaurant(reasturant_code, reasturant_name)

OrderItem(dish_code(PK/FK),order unit,dishRate,line_total)
Order(SN(PK),date_time,order_number,order_amount,delivery_point,status)
OrderRegisterTable(SN(PK/FK),dishCode(PK/FK)

3.ER- MODEL:

3.1. Assumptions:

i. Dishes manages the relation between dish_item and restaurant where one restaurant can have many dishes and one dish_item can be in many restaurants.

ii. OrderRegisterTable handles the relation between order and items that were ordered. A single user can order many items in a 1 order so there exists one to many relationships.

iii. OrderItem contains many dishes_items which is related using dish_code

iv. One Restaurant can have many dish_item.

v. One restaurant can have many dishes and many dishes can contain the same dish item.

vi. One Order can have many order items.

vii. This relation is set up using Order_OrderItem_.

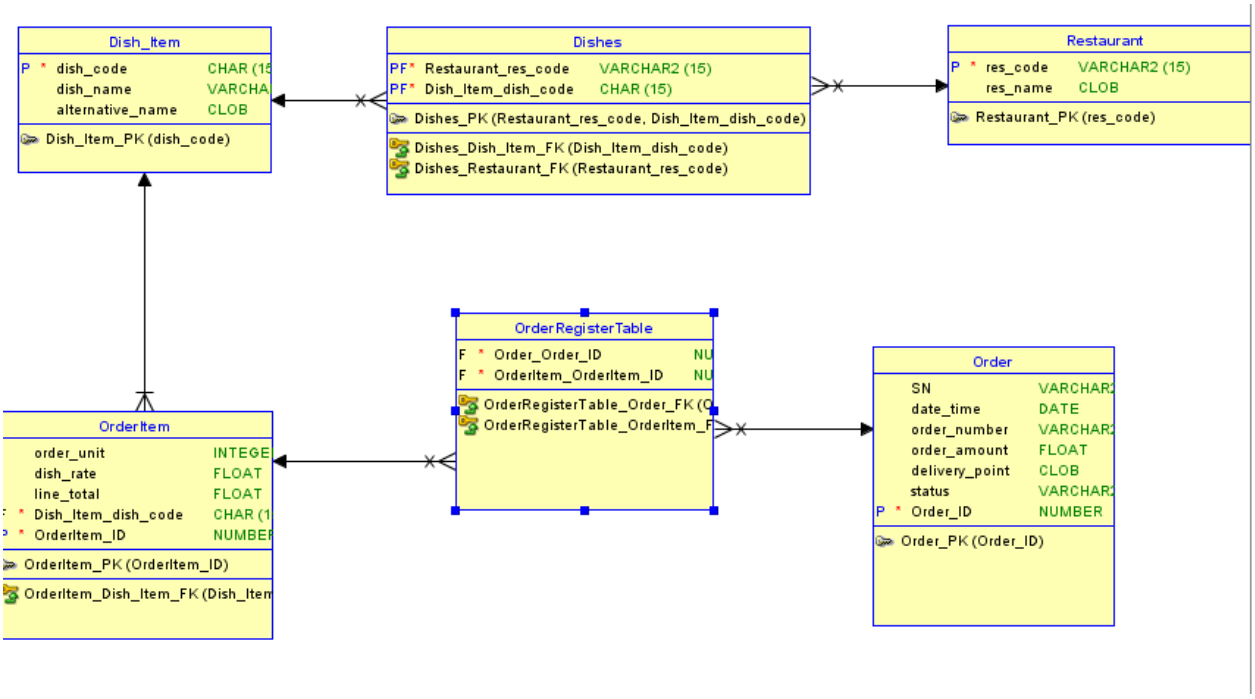


Figure 2: figure of entity relationship diagram

4. Data Dictionary:

Dish Item

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DISH_CODE	CHAR(15 BYTE)	No	(null)	1	(null)
2	DISH_NAME	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	ALTERNATIVE_NAME	CLOB	Yes	(null)	3	(null)

Dishes

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RESTAURANT_RES_CODE	VARCHAR2(15 BYTE)	No	(null)	1	(null)
2	DISH_ITEM_DISH_CODE	CHAR(15 BYTE)	No	(null)	2	(null)

OrderItem

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ORDER_UNIT	NUMBER(38,0)	Yes	(null)	1	(null)
2	DISH_RATE	FLOAT	Yes	(null)	2	(null)
3	LINE_TOTAL	FLOAT	Yes	(null)	3	(null)
4	DISH_ITEM_DISH_CODE	CHAR(15 BYTE)	No	(null)	4	(null)
5	ORDERITEM_ID	NUMBER	No	(null)	5	(null)

Order Register Table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ORDERS_ORDERS_ID	NUMBER	No	(null)	1	(null)
2	ORDERITEM_ORDERITEM_ID	NUMBER	No	(null)	2	(null)

Orders

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SN	VARCHAR2(100 BYTE)	Yes	(null)	1	(null)
2	DATE_TIME	DATE	Yes	(null)	2	(null)
3	ORDER_NUMBER	VARCHAR2(100 BYTE)	Yes	(null)	3	(null)
4	ORDER_AMOUNT	FLOAT	Yes	(null)	4	(null)
5	DELIVERY_POINT	CLOB	Yes	(null)	5	(null)
6	STATUS	VARCHAR2(100 BYTE)	Yes	(null)	6	(null)
7	ORDERS_ID	NUMBER	No	(null)	7	(null)

Restaurant

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RES_CODE	VARCHAR2(15 BYTE)	No	(null)	1	(null)
2	RES_NAME	CLOB	Yes	(null)	2	(null)

5. Generation of Database:

5.1. Create Statement:

```
CREATE TABLE dish_item (
dish_code        CHAR(15) NOT NULL,
dish_name        VARCHAR2(100),
alternative_name  CLOB
);
ALTER TABLE dish_item ADD CONSTRAINT dish_item_pk PRIMARY KEY ( dish_code );
```

```
CREATE TABLE dish_item (
    dish_code         CHAR(15) NOT NULL,
    dish_name         VARCHAR2(100),
    alternative_name  CLOB
);

ALTER TABLE dish_item ADD CONSTRAINT dish_item_pk PRIMARY KEY ( dish_code );
```

Figure 3: figure of creation of dish_item table

```
CREATE TABLE dishes (
   restaurant_res_code  VARCHAR2(15) NOT NULL,
   dish_item_dish_code  CHAR(15) NOT NULL
);

ALTER TABLE dishes ADD CONSTRAINT dishes_pk PRIMARY KEY ( restaurant_res_code,
dish_item_dish_code );
```

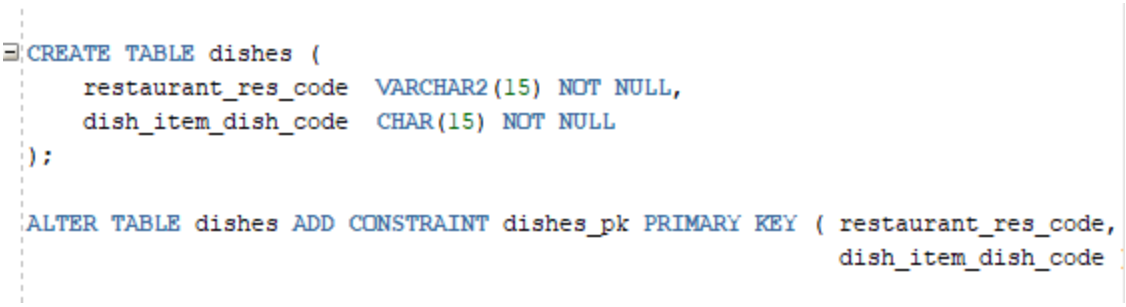


Figure 4: figure of creation of dishes table

```
CREATE TABLE orderitem (
   order_unit         INTEGER,
   dish_rate          FLOAT,
   line_total         FLOAT,
   dish_item_dish_code  CHAR(15) NOT NULL,
   orderitem_id        NUMBER NOT NULL
);

ALTER TABLE orderitem ADD CONSTRAINT orderitem_pk PRIMARY KEY ( orderitem_id );
```

```
CREATE TABLE orderitem (
    order_unit           INTEGER,
    dish_rate            FLOAT,
    line_total           FLOAT,
    dish_item_dish_code  CHAR(15) NOT NULL,
    orderitem_id         NUMBER NOT NULL
);

ALTER TABLE orderitem ADD CONSTRAINT orderitem_pk PRIMARY KEY ( orderitem_id );
```

Figure 5: figure of creation of orderitem

```
CREATE TABLE orderregistertable (
   orders_orders_id       NUMBER NOT NULL,
   orderitem_orderitem_id  NUMBER NOT NULL
);
```

```
CREATE TABLE orderregistertable (
    orders_orders_id        NUMBER NOT NULL,
    orderitem_orderitem_id  NUMBER NOT NULL
);
```

Figure 6: figure of creation of orderregistertable table

```
CREATE TABLE orders (
   sn           VARCHAR2(100),
   date_time    DATE,
   order_number   VARCHAR2(100),
   order_amount   FLOAT,
   delivery_point  CLOB,
   status       VARCHAR2(100),
   orders_id    NUMBER NOT NULL
);

ALTER TABLE orders ADD CONSTRAINT orders_pk PRIMARY KEY ( orders_id );
```

```
CREATE TABLE orders (
    sn              VARCHAR2(100),
    date_time       DATE,
    order_number    VARCHAR2(100),
    order_amount    FLOAT,
    delivery_point  CLOB,
    status          VARCHAR2(100),
    orders_id       NUMBER NOT NULL
);
```

Figure 7: figure of creation of orders table

```
CREATE TABLE restaurant (
    res_code  VARCHAR2(15) NOT NULL,
    res_name  CLOB
);
```

Figure 8: figure of creation of restaurant

5.2. Insert Statement:

Dish Item

```
INSERT ALL
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('CHKMO','Chicken Momo',
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('VEGTM','Vegetable Tem
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('MIXCHT','Mixed Chatam
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('ALURT','ALU Roti','Alu
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('RTOVL','Red Chlly','Ra
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('KHANE','Khanea hoo','K
    INTO dish_item(dish_code,dish_name,alternative_name) VALUES ('CHIKG','Chicken Leg P
SELECT 1 from dual;
```

Figure 9: figure of insertion into dish_item

	DISH_CODE	DISH_NAME	ALTERNATIVE_NAME
1	CHKMO	Chicken Momo	Chicken Momo
2	VEGTM	Vegetable Tempura	Vegetable Tempura
3	MIXCHT	Mixed Chatamari	Nepali Pizza
4	ALURT	ALU Roti	Alu Roti
5	RTOVL	Red Chlly	Rato Kursani ko achar
6	KHANE	Khanea hoo	Khanea hoo
7	CHIKG	Chicken Leg Piece	Chicekn Leg Piece

Restaurant

```
INSERT ALL
    INTO restaurant(res_code,res_name) VALUES ('RXL51','Royal Luxury')
    INTO restaurant(res_code,res_name) VALUES ('TAB49','Taste Buds')
    INTO restaurant(res_code,res_name) VALUES ('LOC59','Local Foodie')
    INTO restaurant(res_code,res_name) VALUES ('SI69','Sainli Ko Batti')
    INTO restaurant(res_code,res_name) VALUES ('AMOG12','Amogg Cafe')
SELECT 1 from dual;
```

Figure 10: figure of insertion into restaurant

	RES_CODE	RES_NAME
1	RXL51	Royal Luxury
2	TAB49	Taste Buds
3	LOC59	Local Foodie
4	SI69	Sainli Ko Batti
5	AMOG12	Amogg Cafe

Dishes

```
INSERT ALL
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('RXL51','CHKMO')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('RXL51','VEGTM')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('RXL51','MIXCHT')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('LOC59','CHKMO')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('LOC59','MIXCHT')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('LOC59','ALURT')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('AMOG12','ALURT')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('AMOG12','RTOVL')
    INTO dishes(RESTAURANT_RES_CODE,DISH_ITEM_DISH_CODE) VALUES ('RXL51','RTOVL')
SELECT 1 from dual;
```

Figure 11: figure of insertion into dishes

	RESTAURANT_RES_CODE	DISH_ITEM_DISH_CODE
1	AMOG12	ALURT
2	AMOG12	RTOVL
3	LOC59	ALURT
4	LOC59	CHKMO
5	LOC59	MIXCHT
6	RXL51	CHKMO
7	RXL51	MIXCHT
8	RXL51	RTOVL
9	RXL51	VEGTM

OrderItem

```
INSERT ALL
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('CHKMO',1,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('VEGTM',2,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('VEGTM',3,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('CHKMO',4,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('CHKMO',5,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('RTOVL',6,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('VEGTM',7,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('CHKMO',8,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('CHKMO',9,10,120,1200)
    INTO orderitem(DISH_ITEM_DISH_CODE,orderitem_id,order_unit,dish_rate,line_total) VALUES ('RTOVL',10,10,120,1200)
SELECT 1 from dual;
```

Figure 12: figure of insertion into orderitem

	ORDER_UNIT	DISH_RATE	LINE_TOTAL	DISH_ITEM_DISH_CODE	ORDERITEM_ID
1	10	120	1200	CHKMO	1
2	10	120	1200	VEGTM	2
3	10	120	1200	VEGTM	3
4	10	120	1200	CHKMO	4
5	10	120	1200	CHKMO	5
6	10	120	1200	RTOVL	6
7	10	120	1200	VEGTM	7
8	10	120	1200	CHKMO	8
9	10	120	1200	CHKMO	9
10	10	120	1200	RTOVL	10

Orders

```
INSERT ALL
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223334',10,250.69,'kumarigal','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223335',10,250.69,'Bouddha','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223336',10,250.69,'bhaktapur','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223337',10,250.69,'banesowore','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223338',10,250.69,'shaku','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-223339',10,250.69,'kumarigal','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-2233310',10,250.69,'bouddha','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-2233311',10,250.69,'udhyapur','ON Time')
    INTO Orders(SN,order_number,order_amount,delivery_point,status) VALUES ('2020-09-2233312',10,250.69,'saknu','ON Time')
SELECT 1 from dual;
```

Figure 13: figure of insertion into orders

	SN	DATE_TIME	ORDER_NUMBER	ORDER_AMOUNT	DELIVERY_P...	STATUS	ORDERS_ID
1	2020-09-223334	(null)	10	250.69	kumarigal	ON Time	1
2	2020-09-223335	(null)	10	250.69	Bouddha	ON Time	2
3	2020-09-223336	(null)	10	250.69	bhaktapur	ON Time	3
4	2020-09-223337	(null)	10	250.69	banesowore	ON Time	4
5	2020-09-223338	(null)	10	250.69	shaku	ON Time	5
6	2020-09-223339	(null)	10	250.69	kumarigal	ON Time	6
7	2020-09-2233310	(null)	10	250.69	bouddha	ON Time	7
8	2020-09-2233311	(null)	10	250.69	udhyapur	ON Time	8
9	2020-09-2233312	(null)	10	250.69	saknu	ON Time	9

OrderRegisterTable

```
INSERT ALL
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(1,1)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(2,2)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(3,3)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(4,4)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(5,5)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(6,6)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(7,7)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(8,8)
    INTO ORDERREGISTERTABLE(ORDERS_ORDERS_ID,ORDERITEM_ORDERITEM_ID) VALUES(9,9)
SELECT 1 from dual;
```

Figure 14: figure of insertion into orderregistertable

	ORDERS_ORDERS_ID	ORDERITEM_ORDERITEM_ID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

6. IMPLEMENTATION OF WEB BASED DATABASE APPLICATION:

6.1. Home Page:

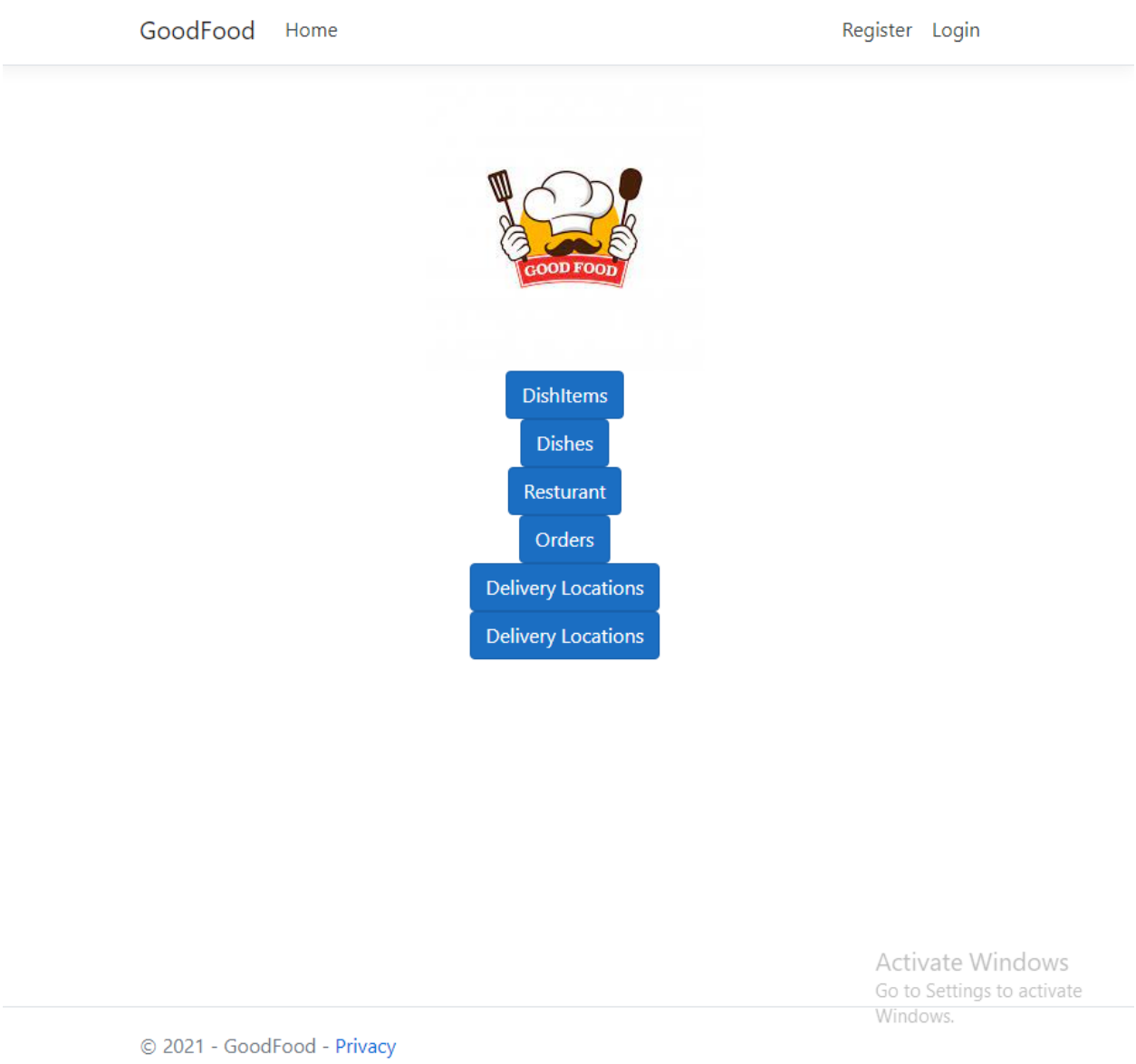


Figure 15: Home page of Webform

6.2. WEBFORMS:

Dish Item

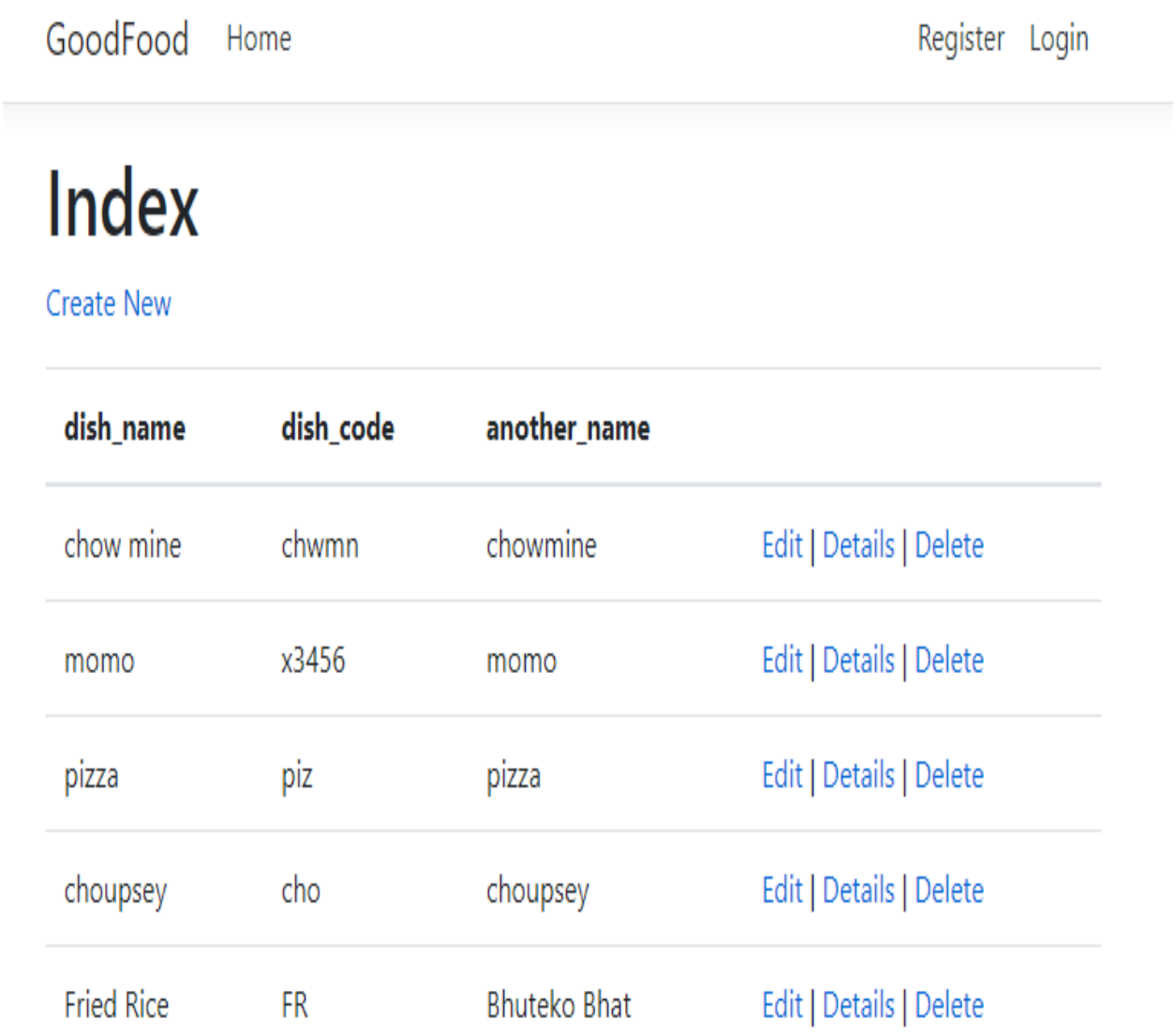


Figure 16: Webform of DishItem

Dishes

Index

Create New

item_rate	dishitem	resturant	
1200	1	1	Edit | Details | Delete
340	2	2	Edit | Details | Delete
1200	2	1	Edit | Details | Delete
450	4	3	Edit | Details | Delete
700	5	3	Edit | Details | Delete

Figure 17:Webform of Dishes

Restaurant

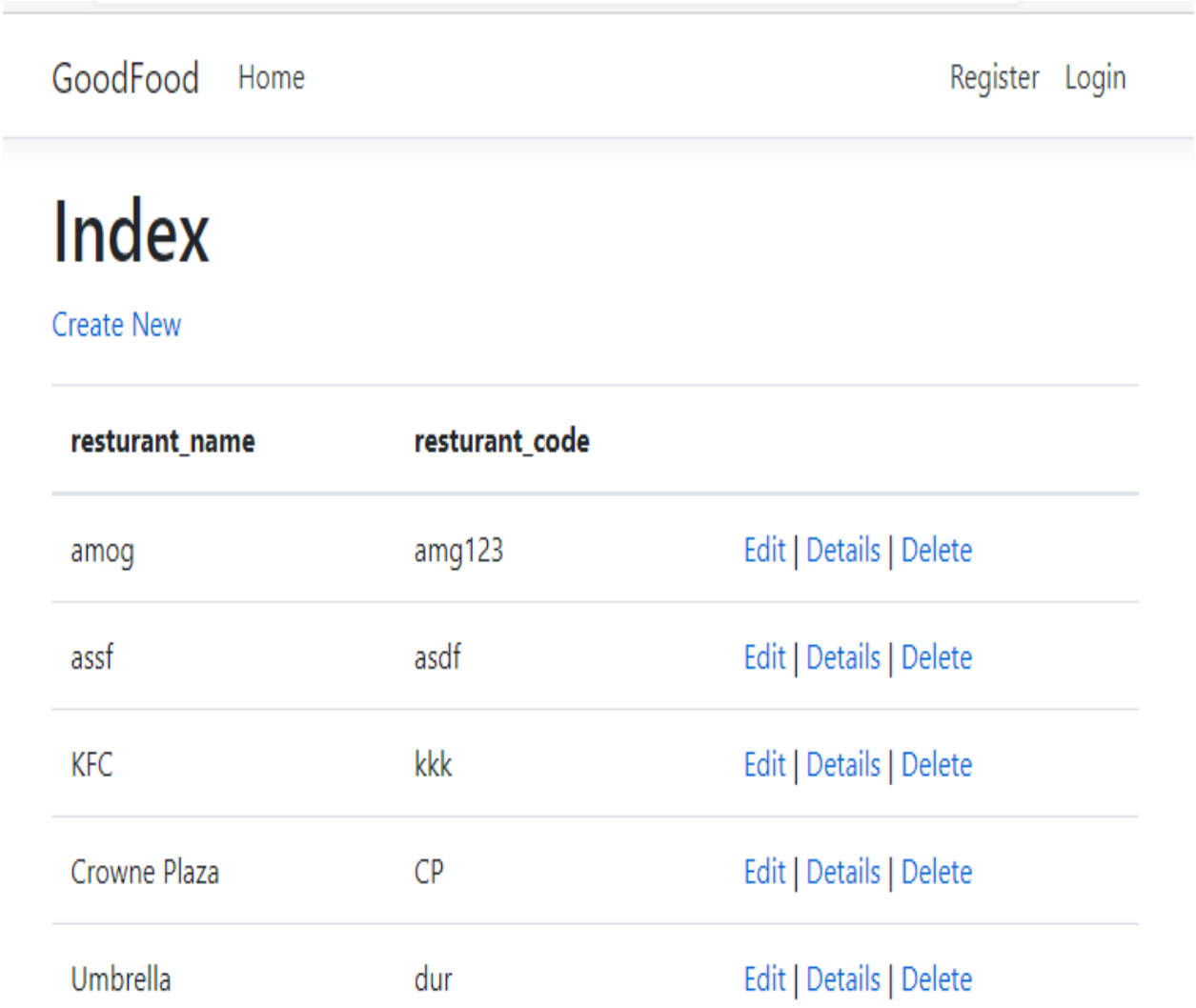


Figure 18:Webform of Restaurant

Orders

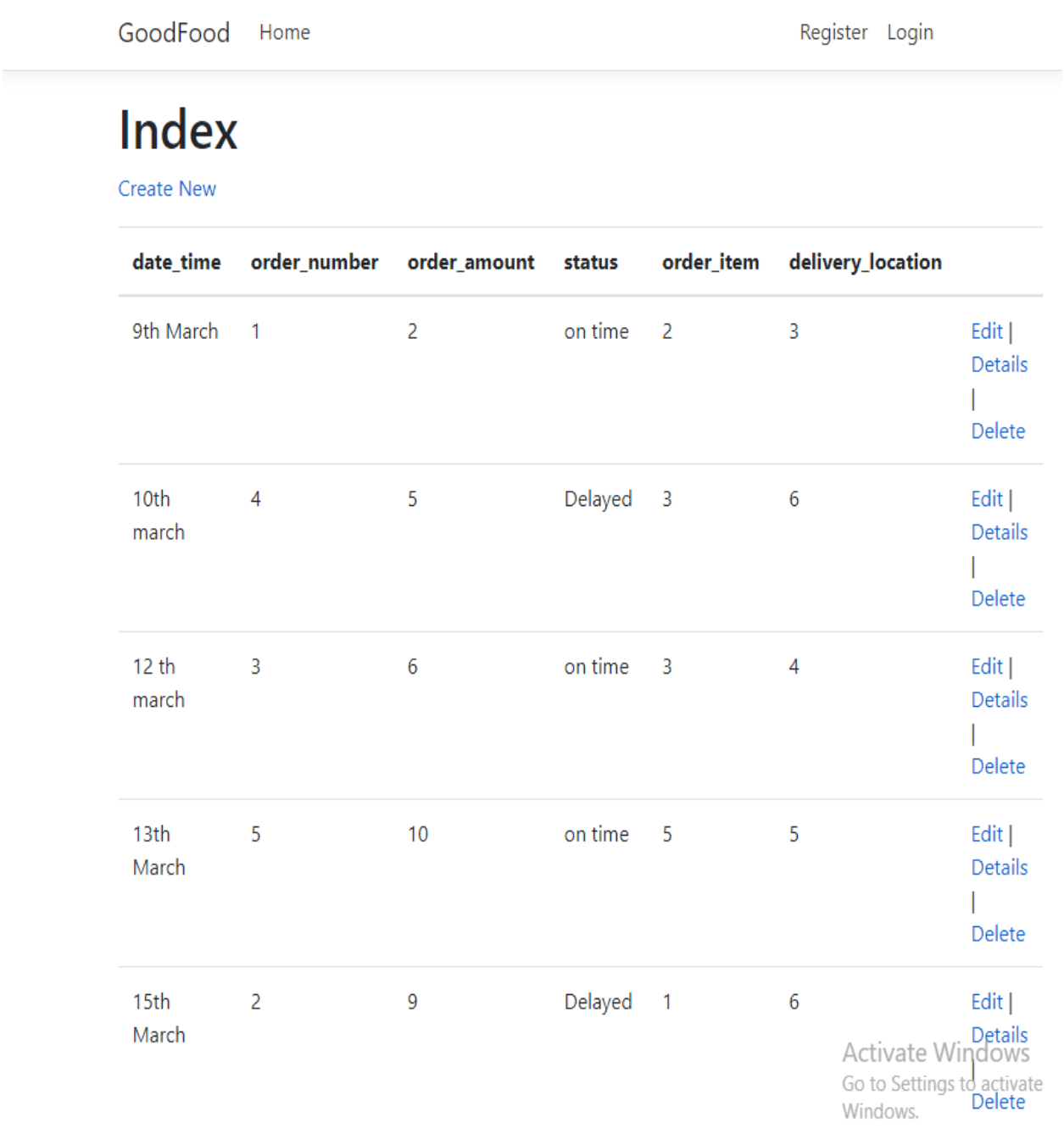


Figure 19:Webform of Orders

Locations

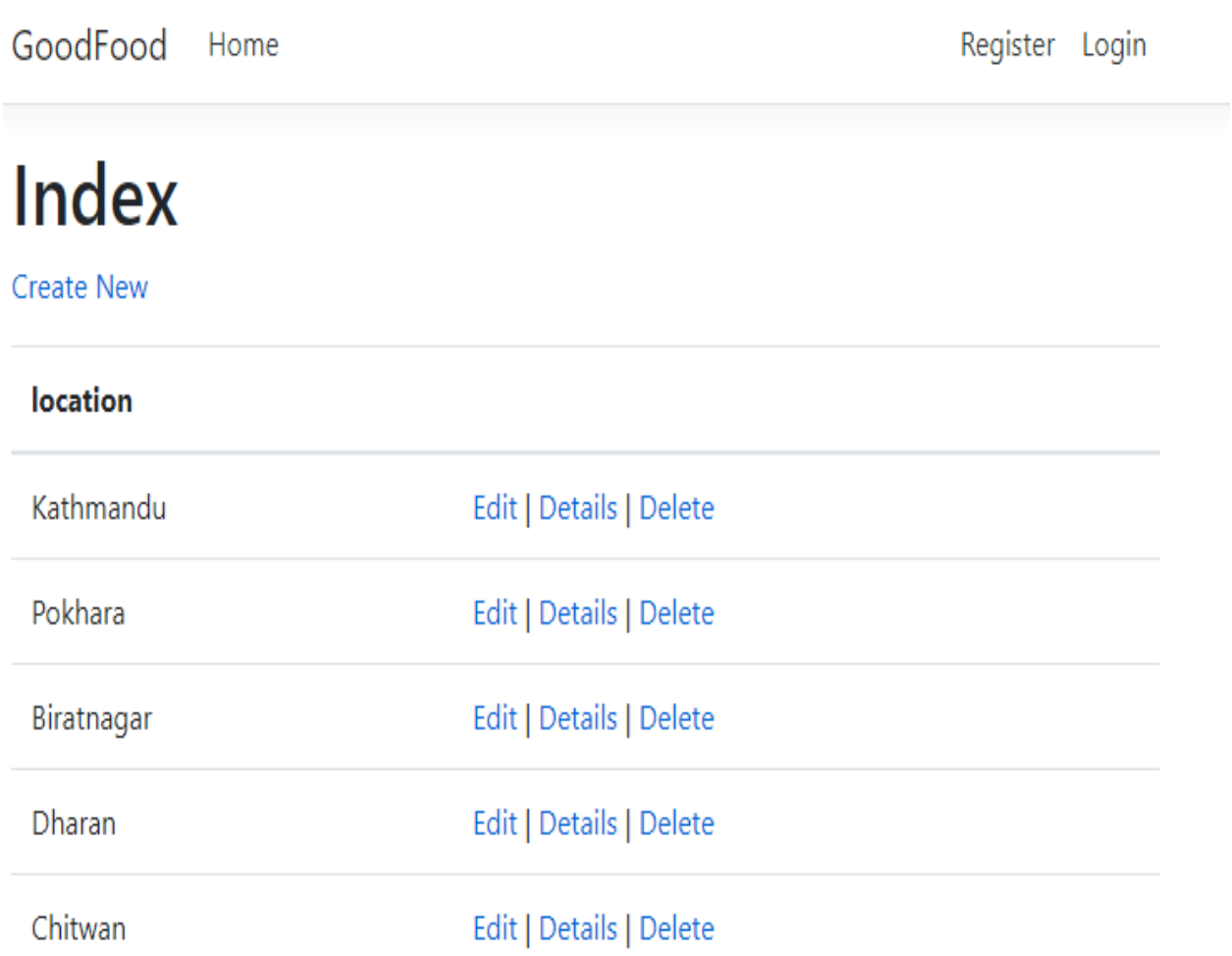


Figure 20:Webform of location

OrderItem

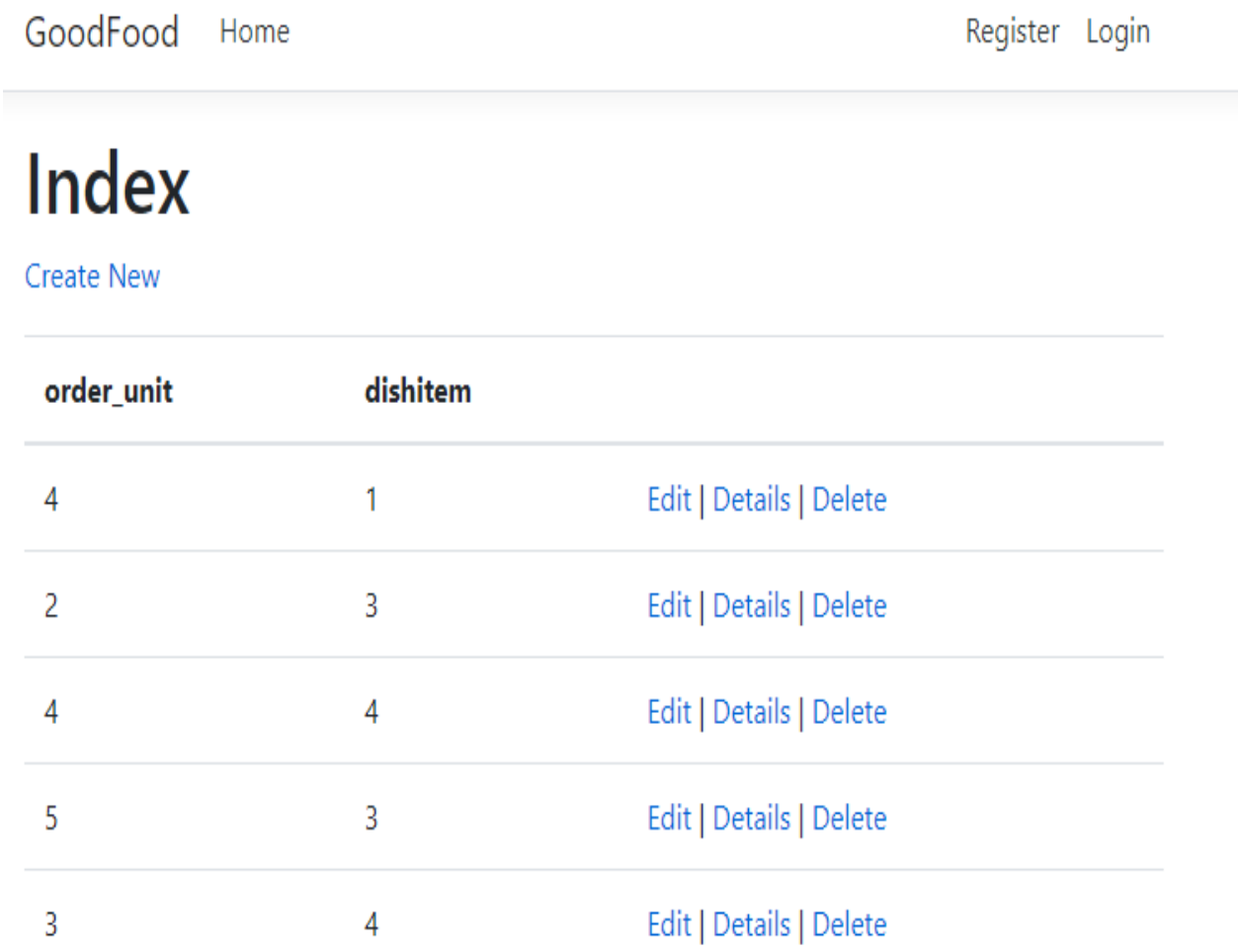


Figure 21:Webform of OrderItem

7. Testing:

7.1. Testing of DishItems:

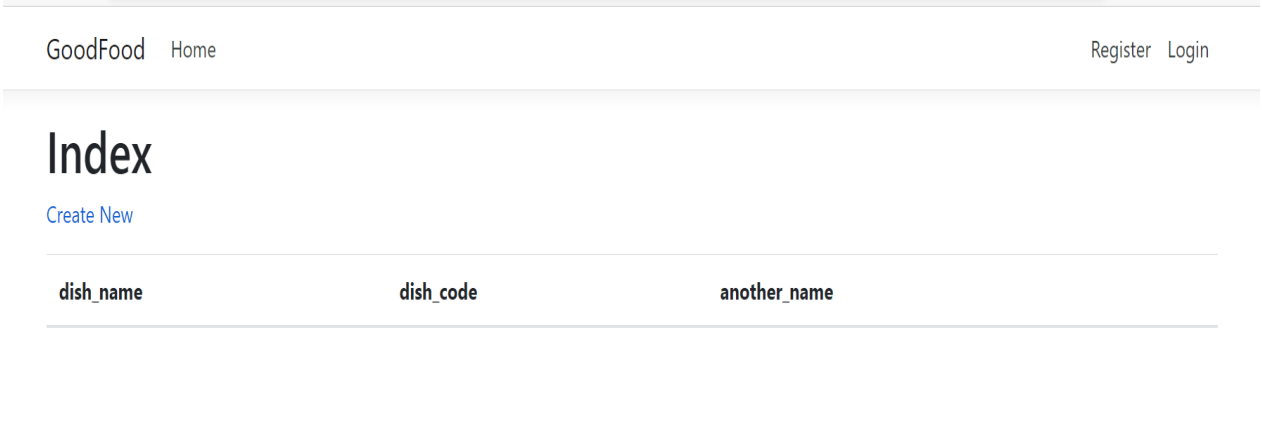


Figure 22: inserting in enplty fields

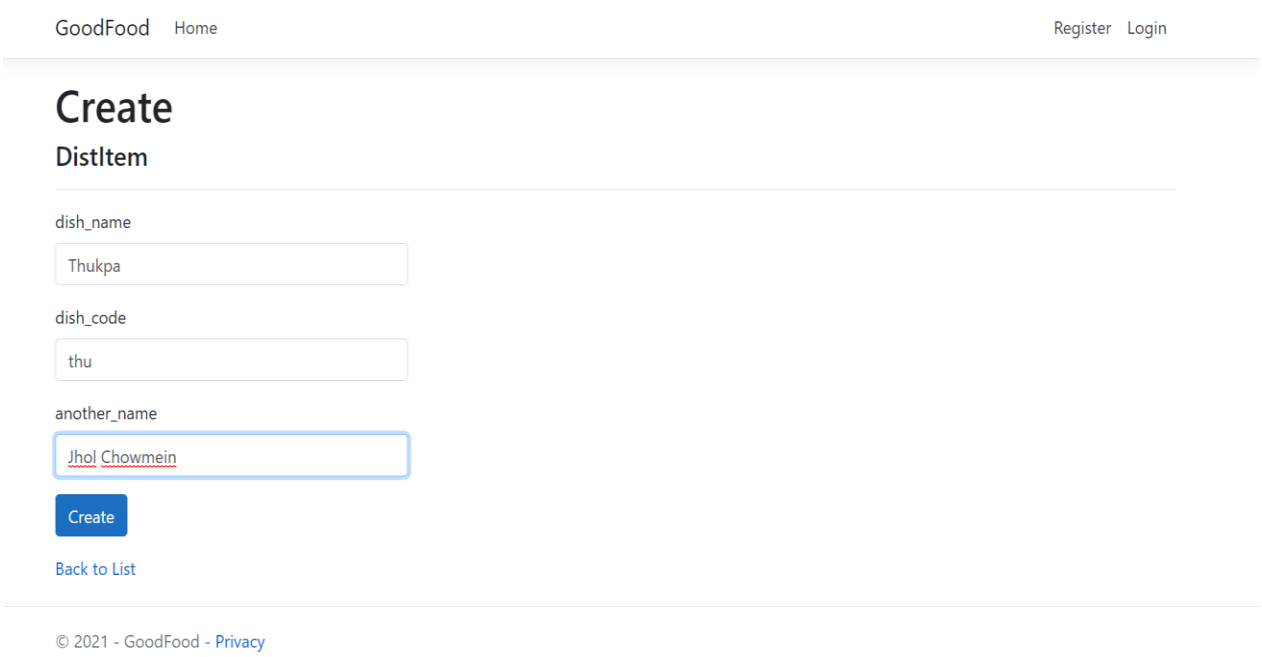


Figure 23:Inserting the datas

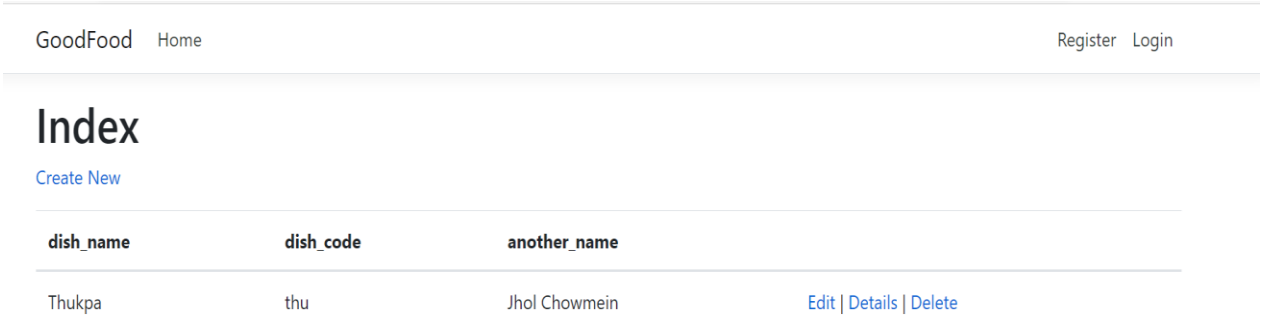


Figure 24:Datas Inserted

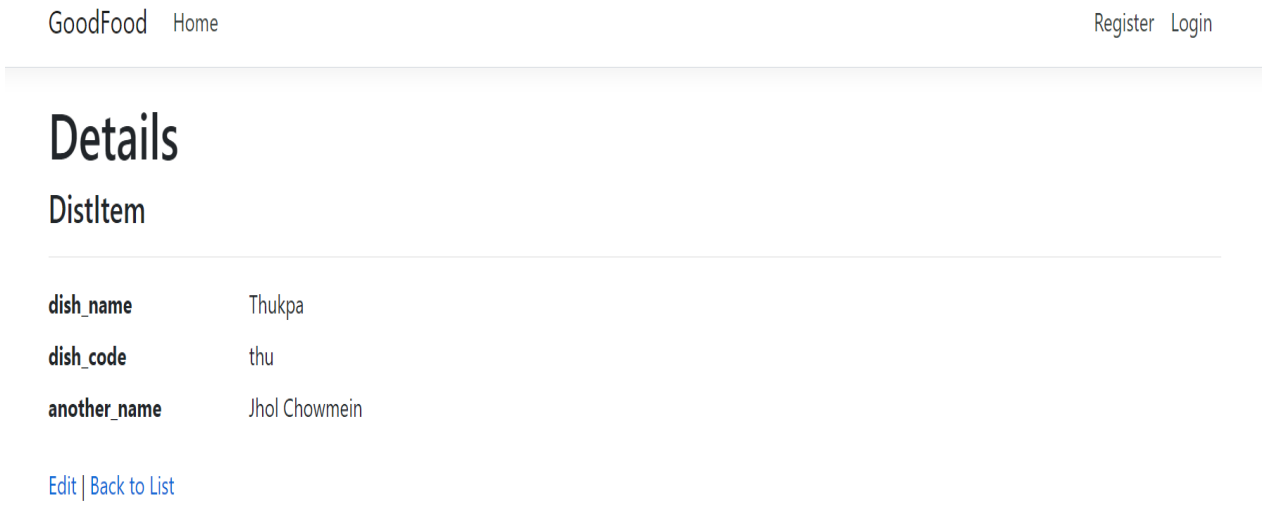


Figure 25:Details of Dish Item

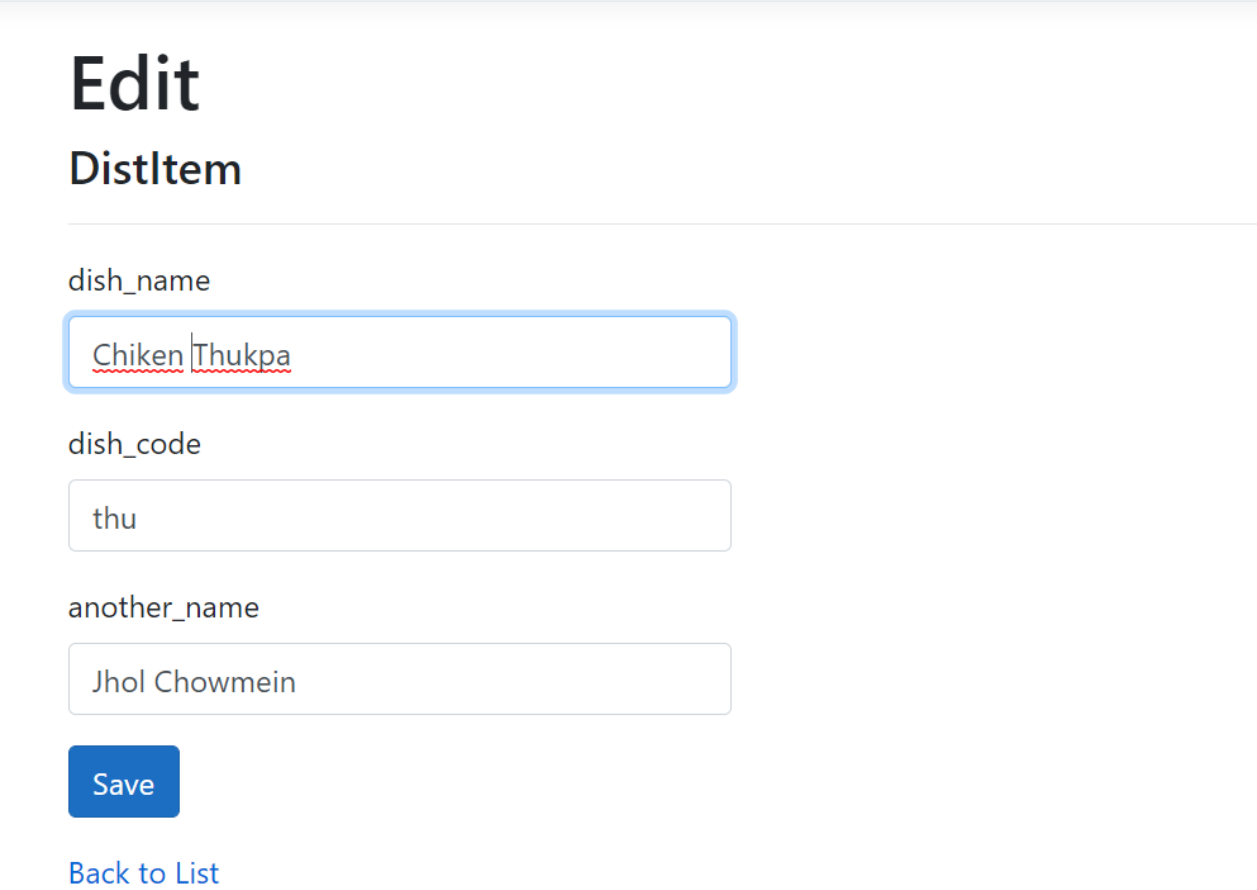


Figure 26:Editing DishItem

Figure 27:Dishitem Edited

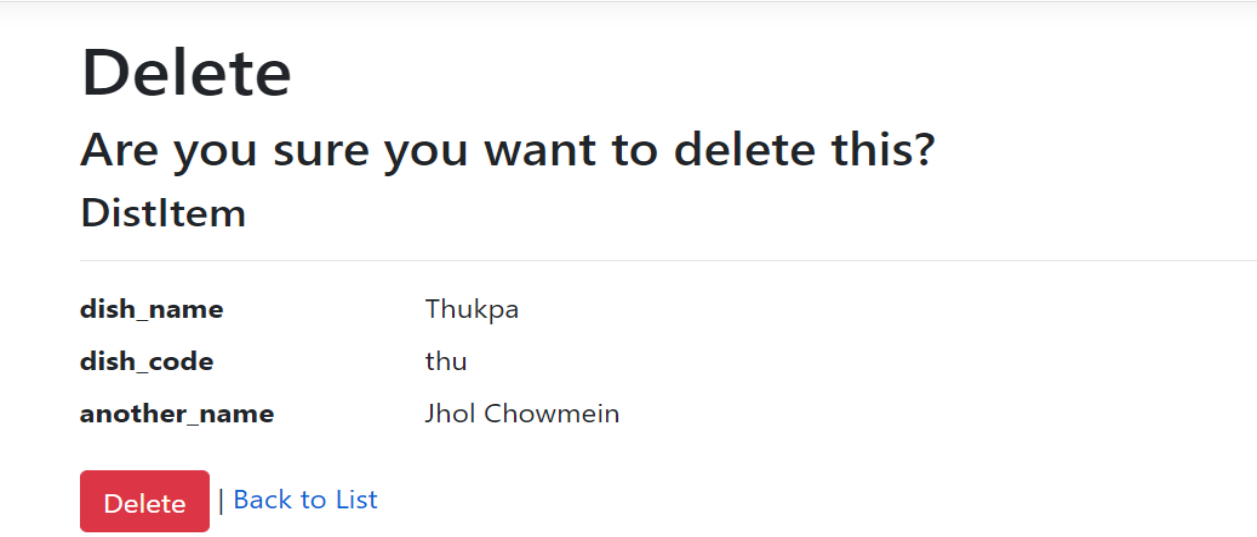


Figure 28:Deleting DishItems

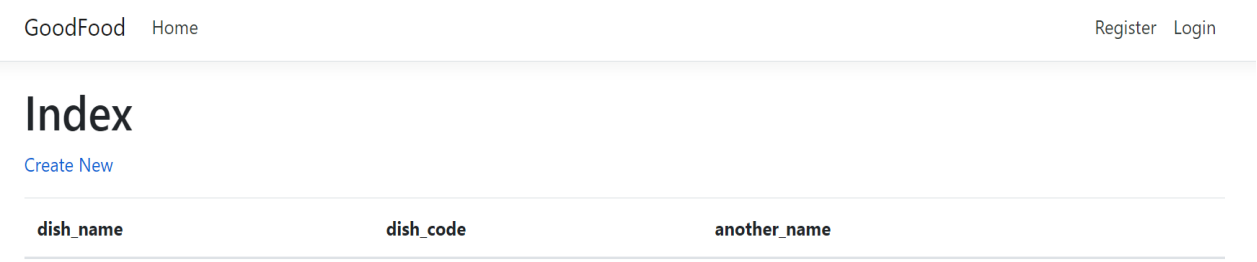


Figure 29:DishItems deleted

Objective	To test the entire webform named DishItems.
Expected Result	To insert, edit and delete data.
Outcome	Data's inserted, edited, and deleted.
Result	Data's inserted, edited, and deleted.

7.2 Testing of Restaurant:

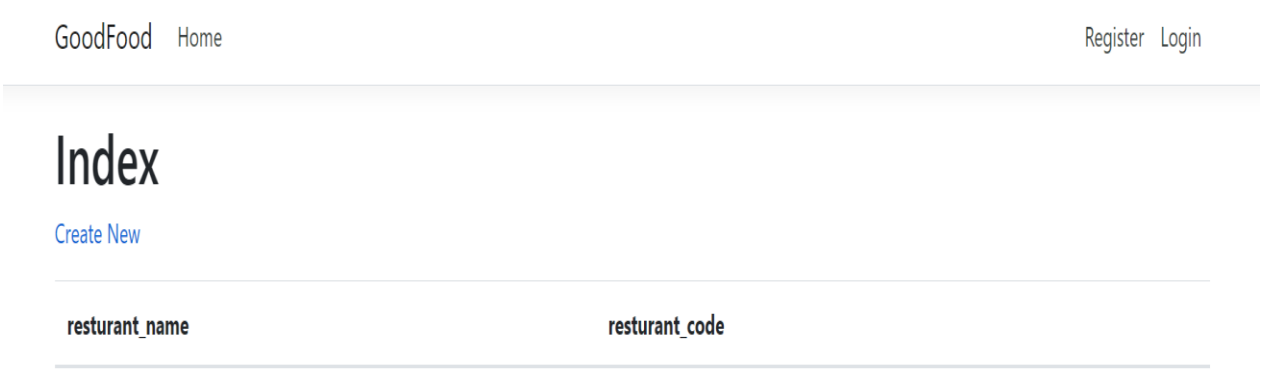


Figure 30: Empty Field Restaurant

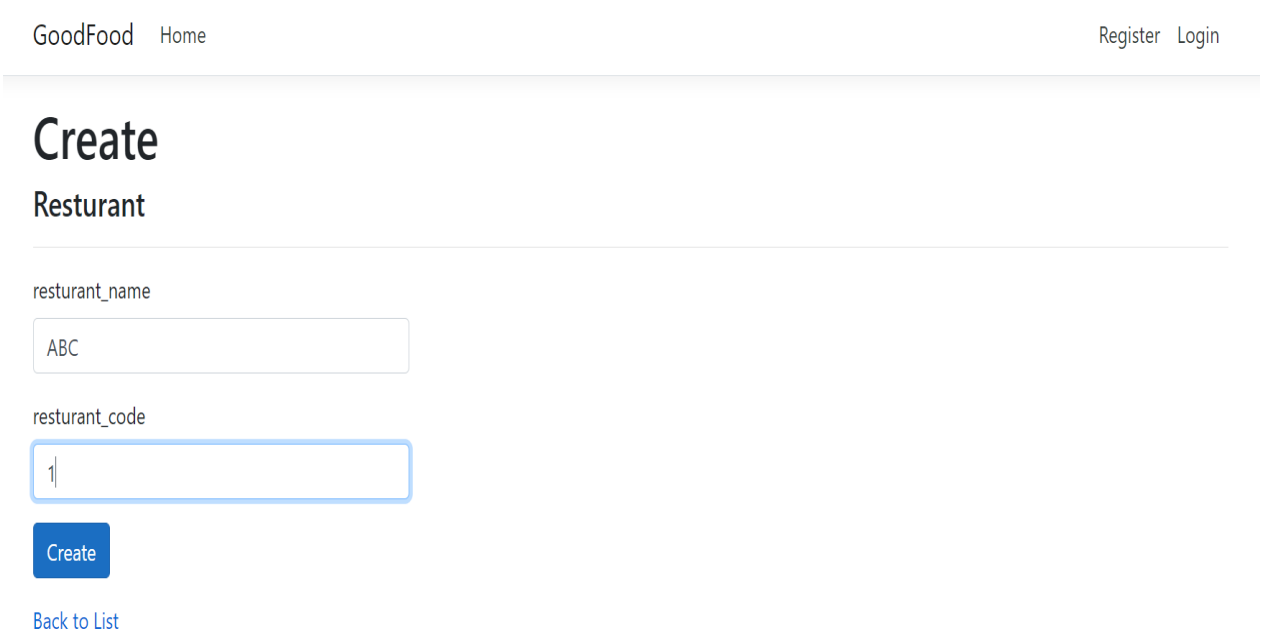


Figure 31:Creating into restaurant

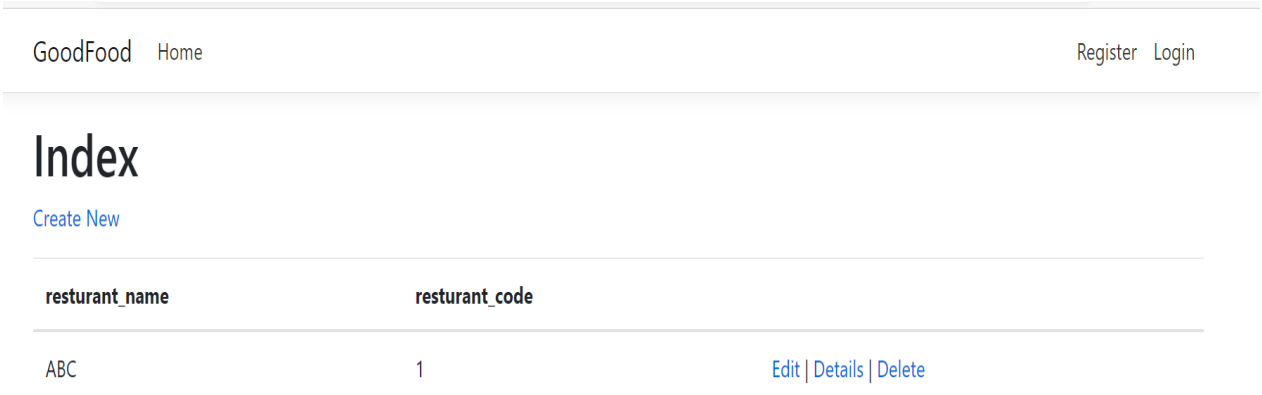


Figure 32:Restaurant Created

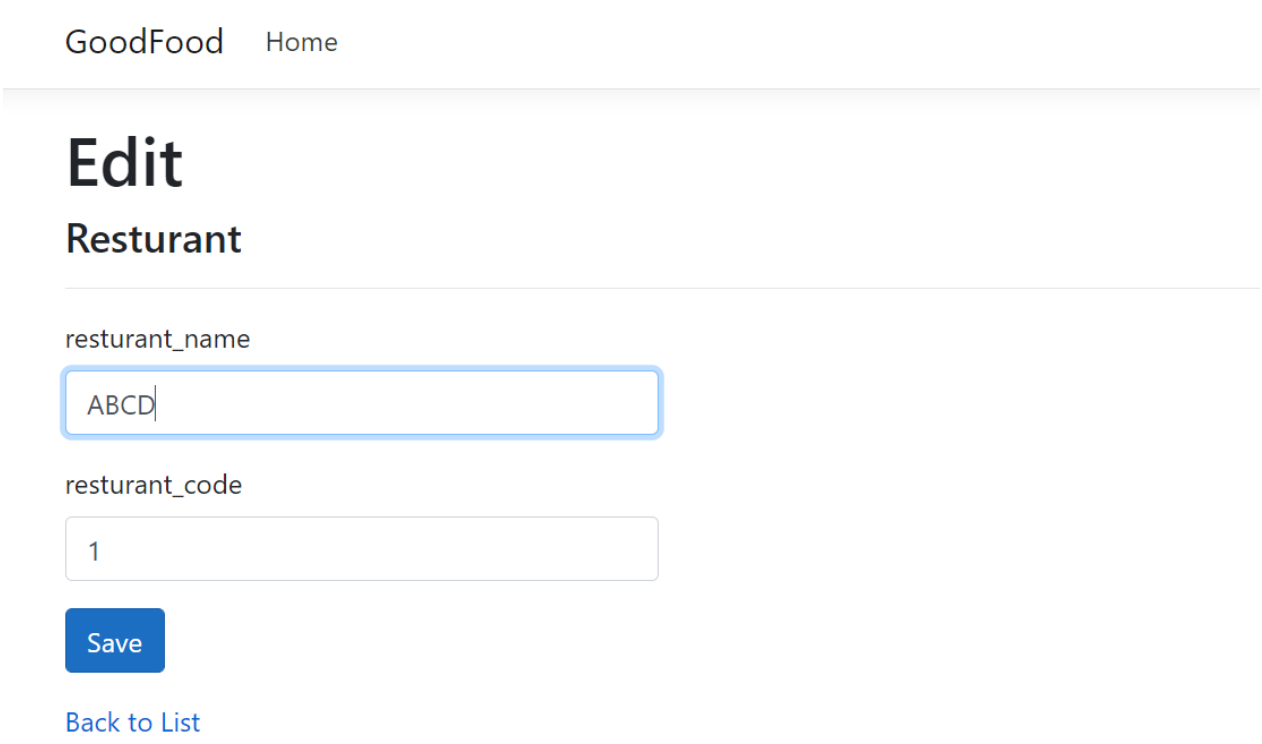


Figure 33:Editing Restaurant

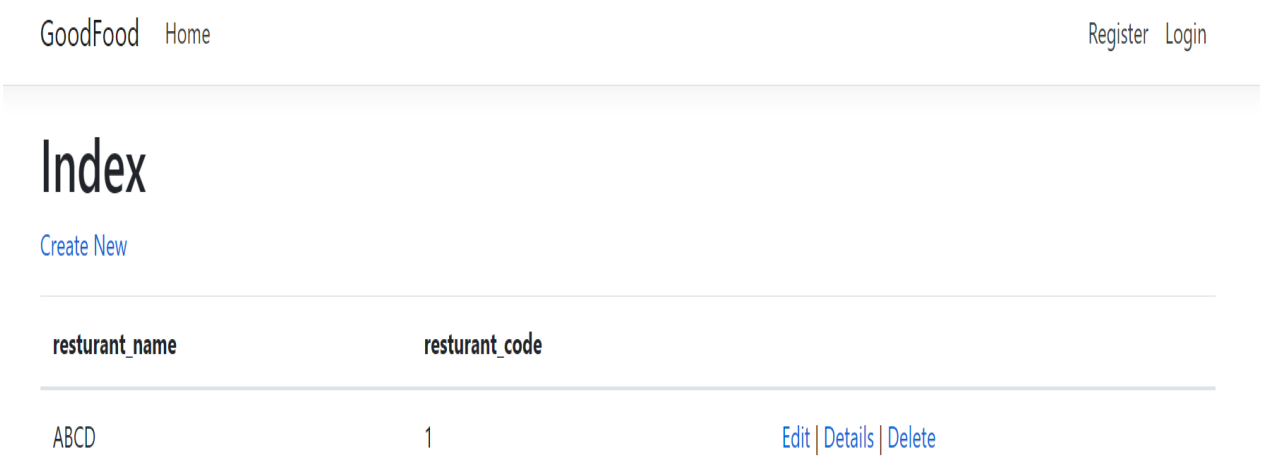


Figure 34:Restaurant Edited

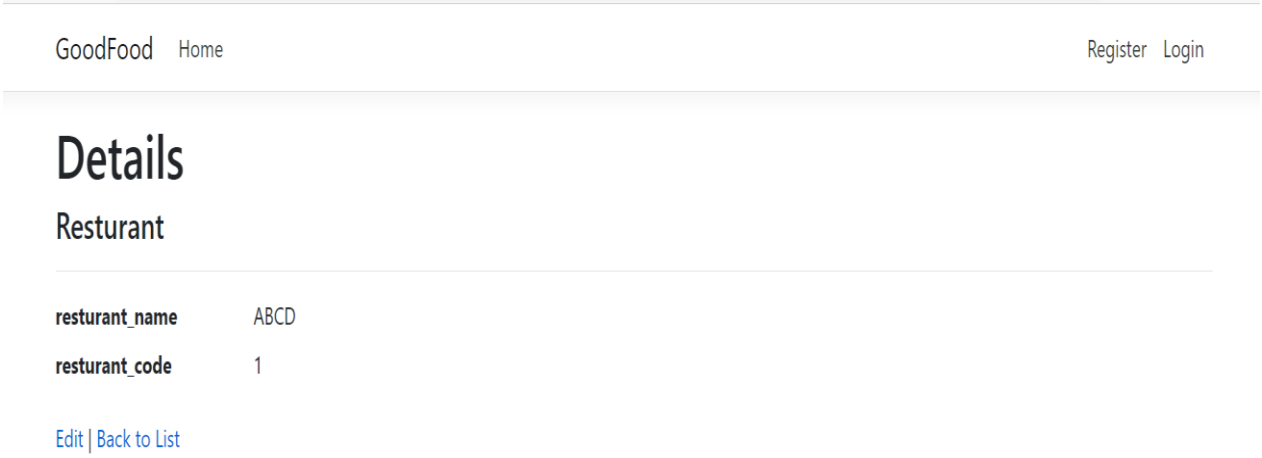


Figure 35:Details of Restaurant

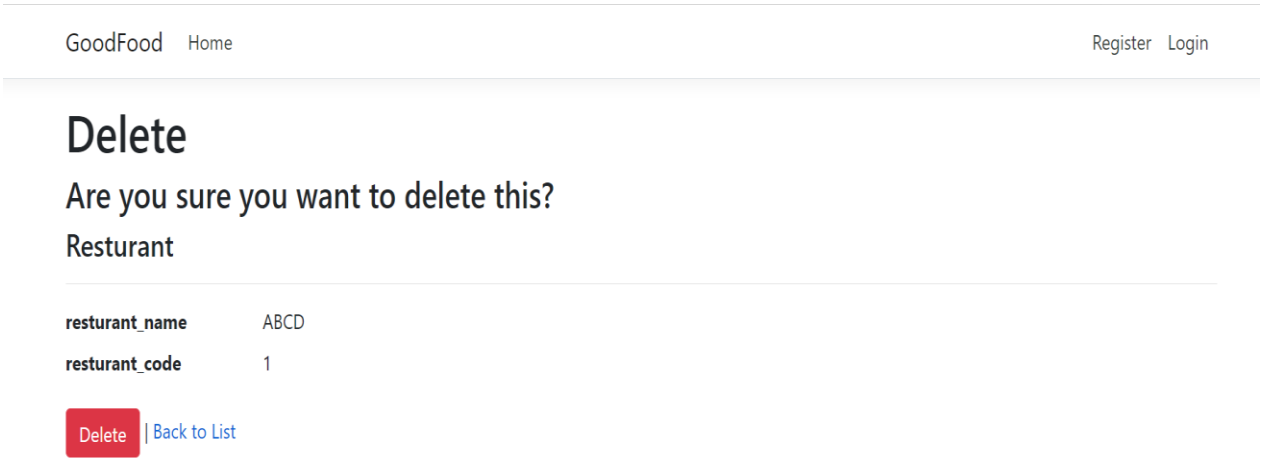


Figure 36:Deleting Restaurant

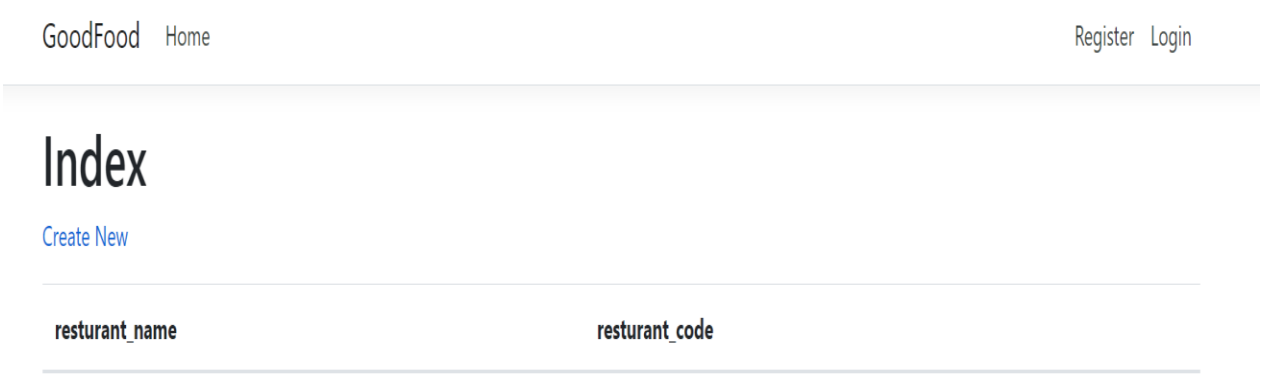


Figure 37:Restaurant Deleted

Objective	To test the entire webform named Restaurant.
Expected Result	To insert, edit and delete data.
Outcome	Data's inserted, edited, and deleted.
Result	Data's inserted, edited, and deleted.

7.3. Testing of Delivery Locations:

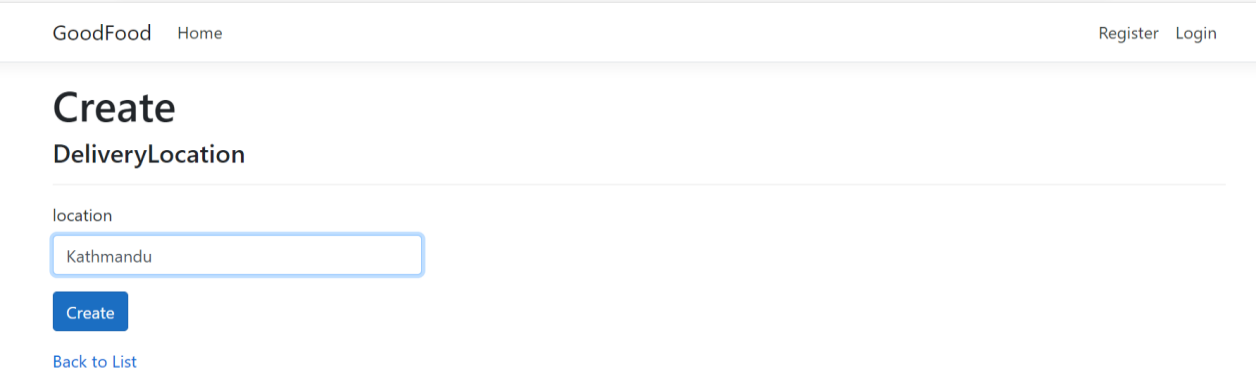


Figure 38:Creating Location

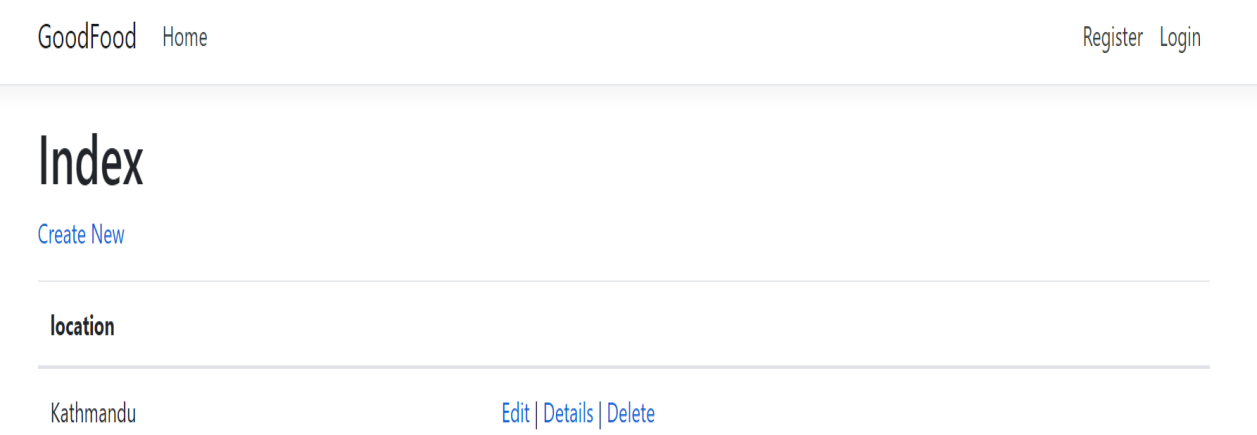


Figure 39:Location Created

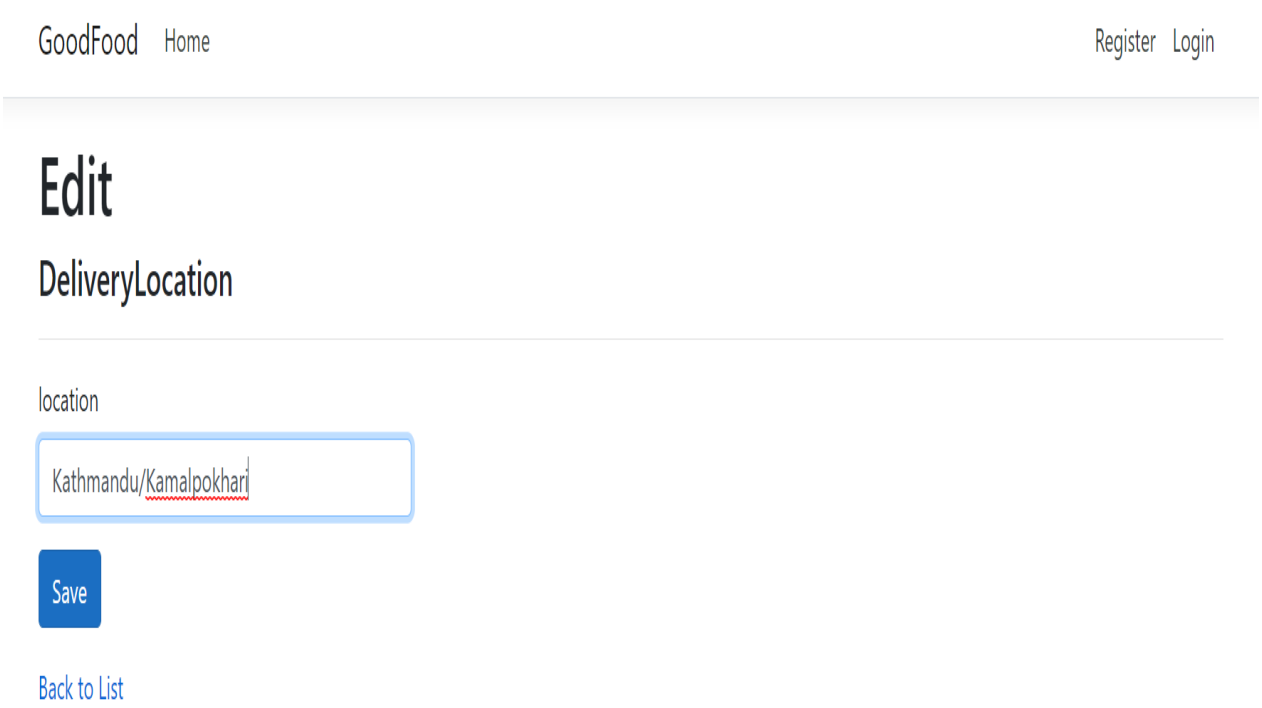


Figure 40:editing Location

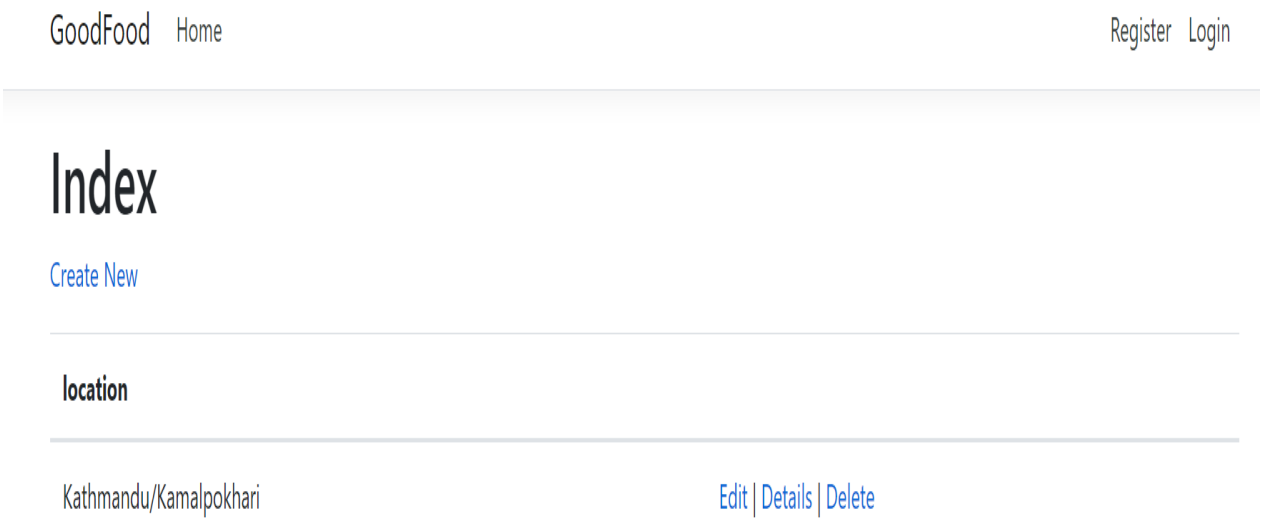


Figure 41:Location Edited

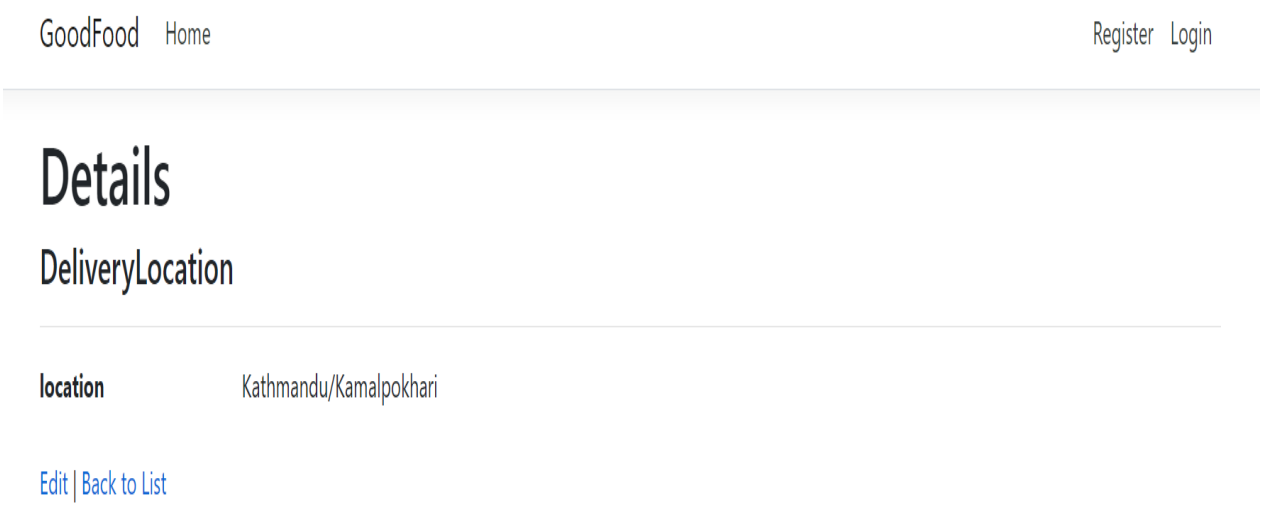


Figure 42:Details of Location

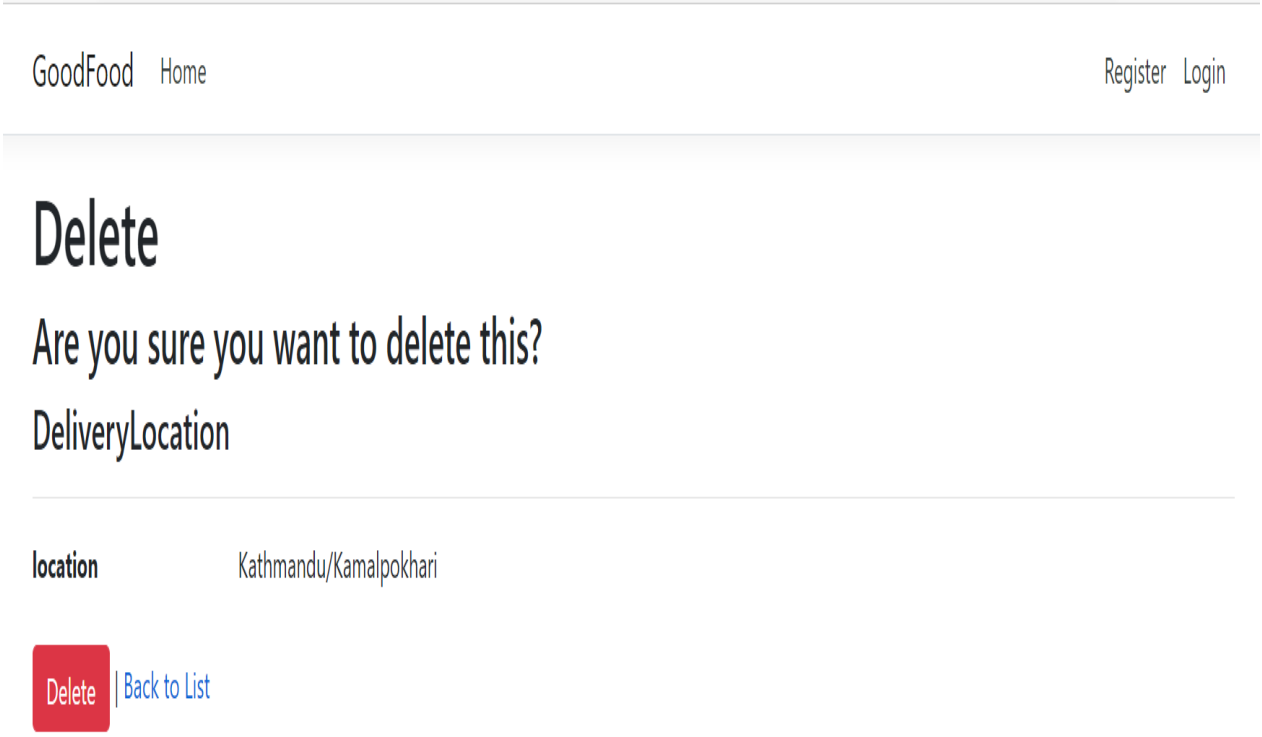


Figure 43:Deleting Location

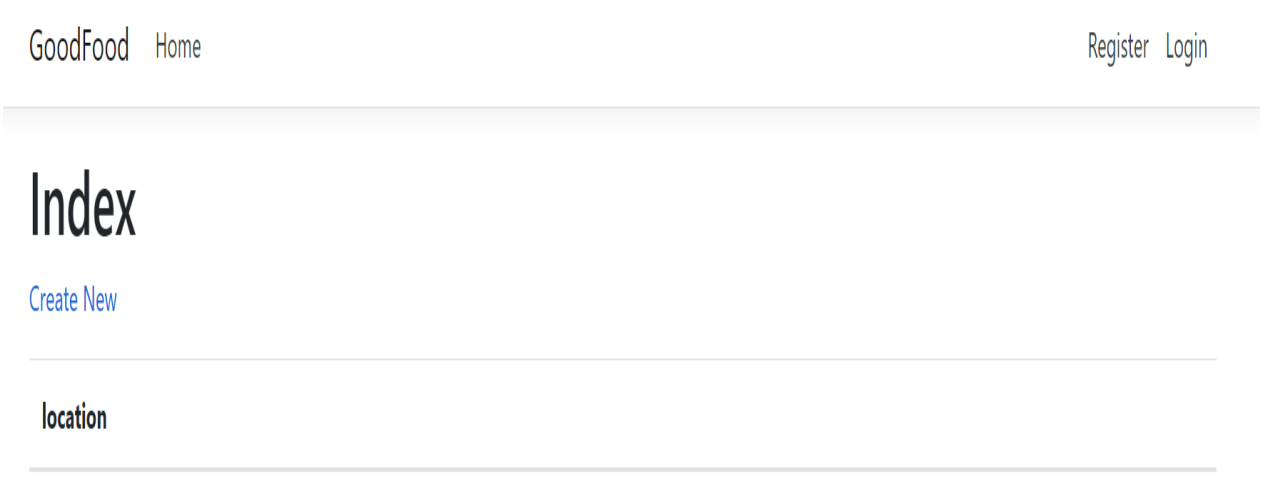


Figure 44:Location Deleted

Objective	To test the entire webform named DeliveryLocation.
Expected Result	To insert, edit and delete data.
Outcome	Data's inserted, edited, and deleted.
Result	Data's inserted, edited, and deleted.

7.4. Testing of OrderItems:

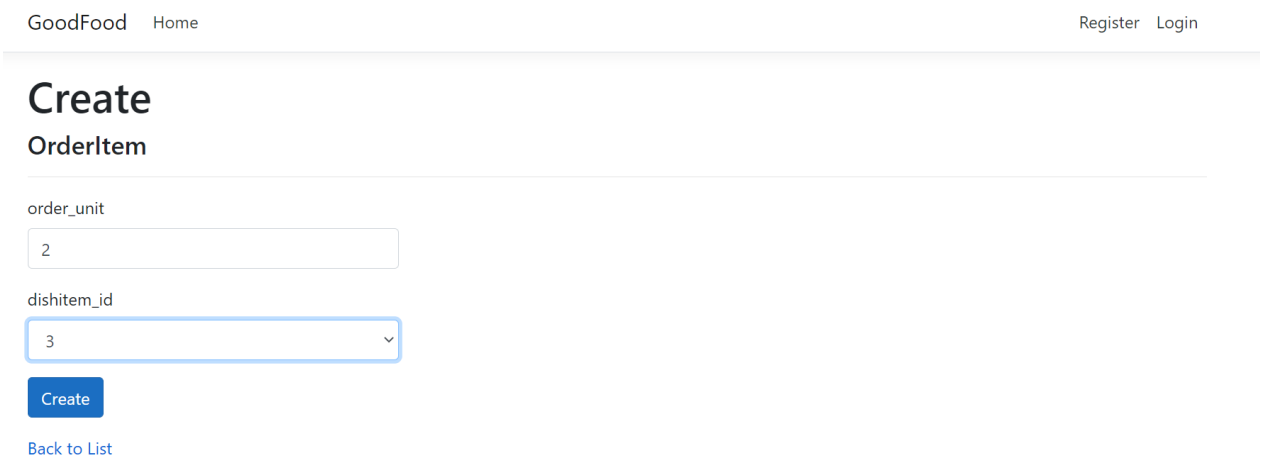


Figure 45:Creating OrderItem

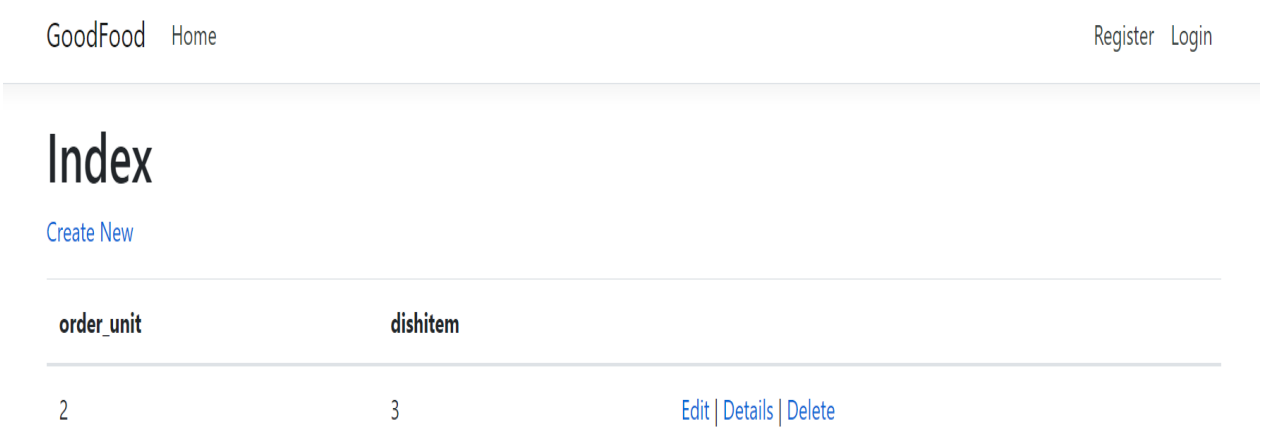


Figure 46:OrderItems Created

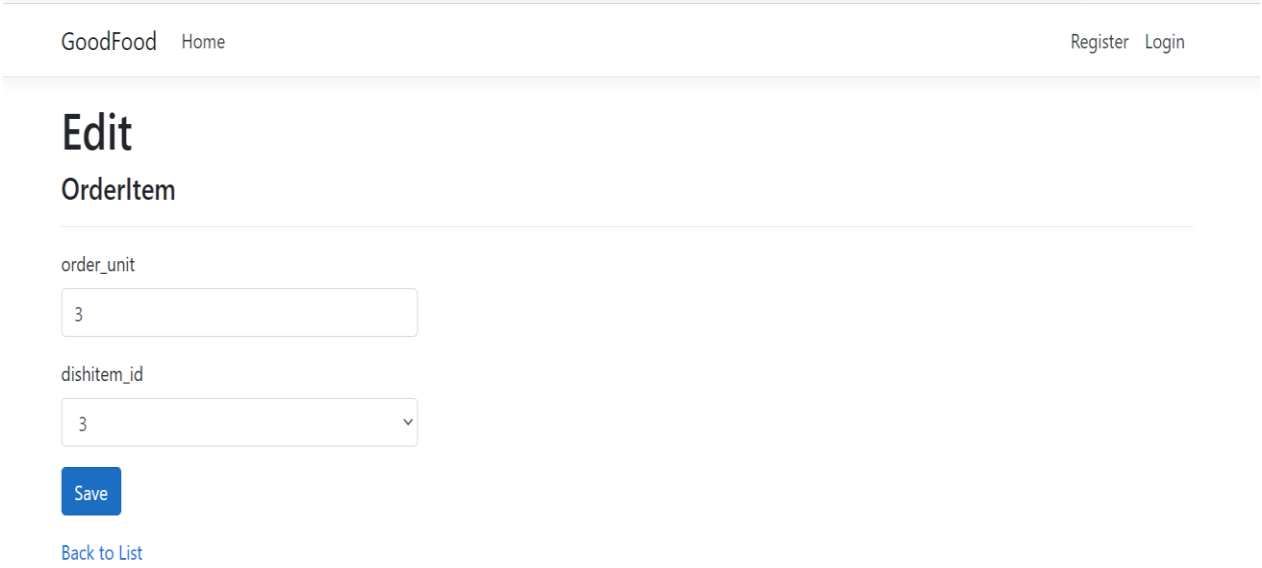


Figure 47:Editing OrderItems

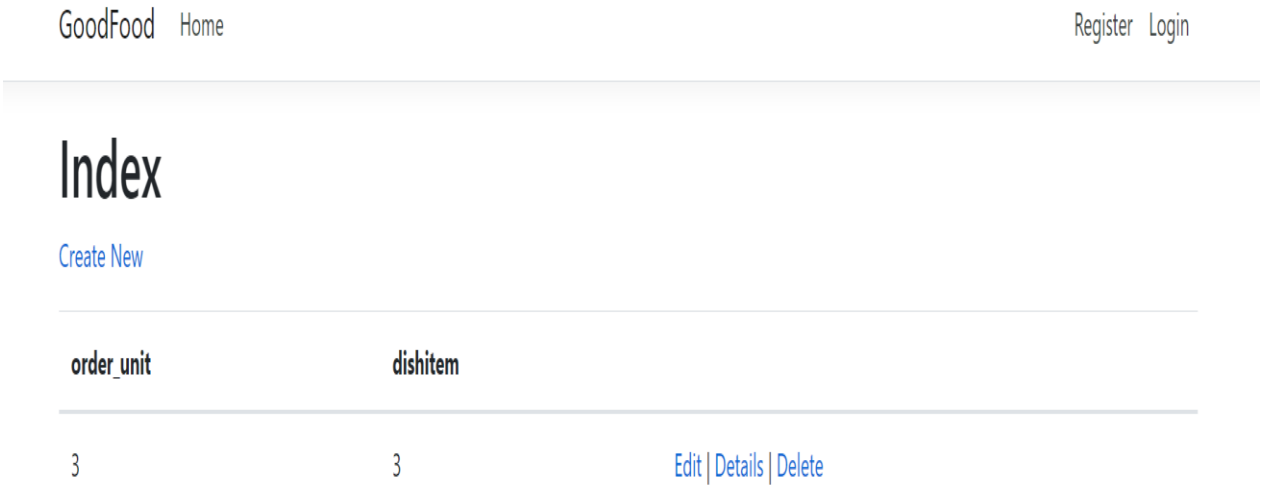


Figure 48:OrderItems Edited

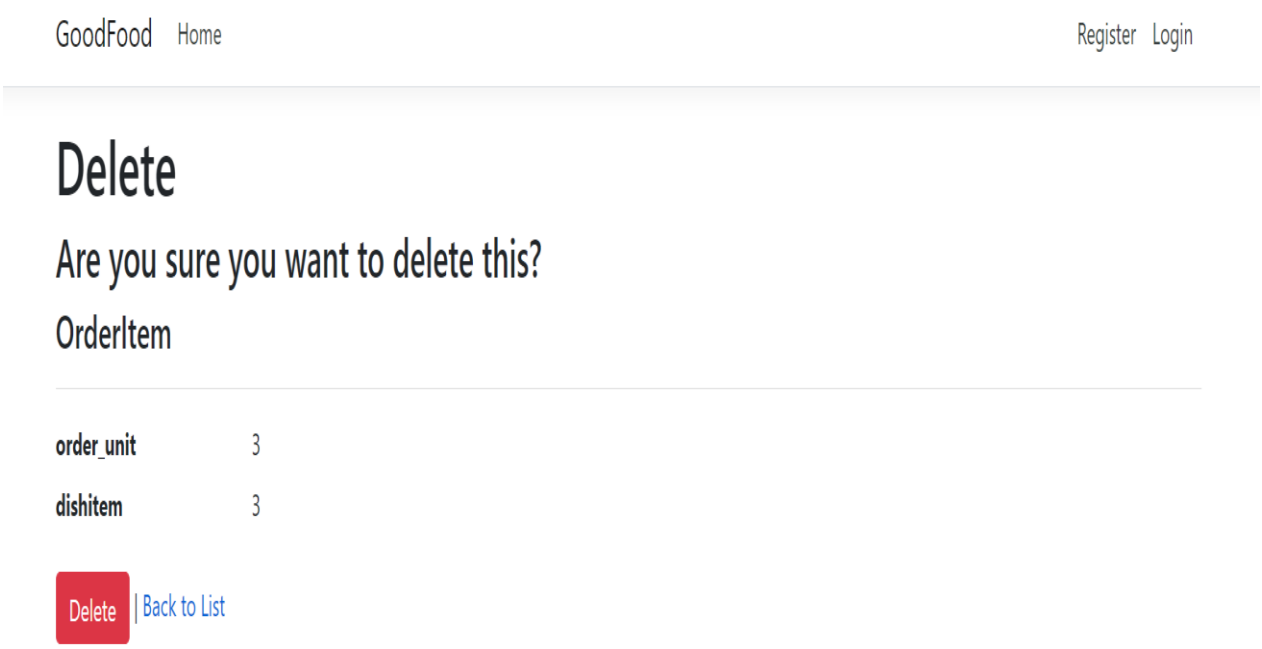


Figure 49:Deleting OrderItems

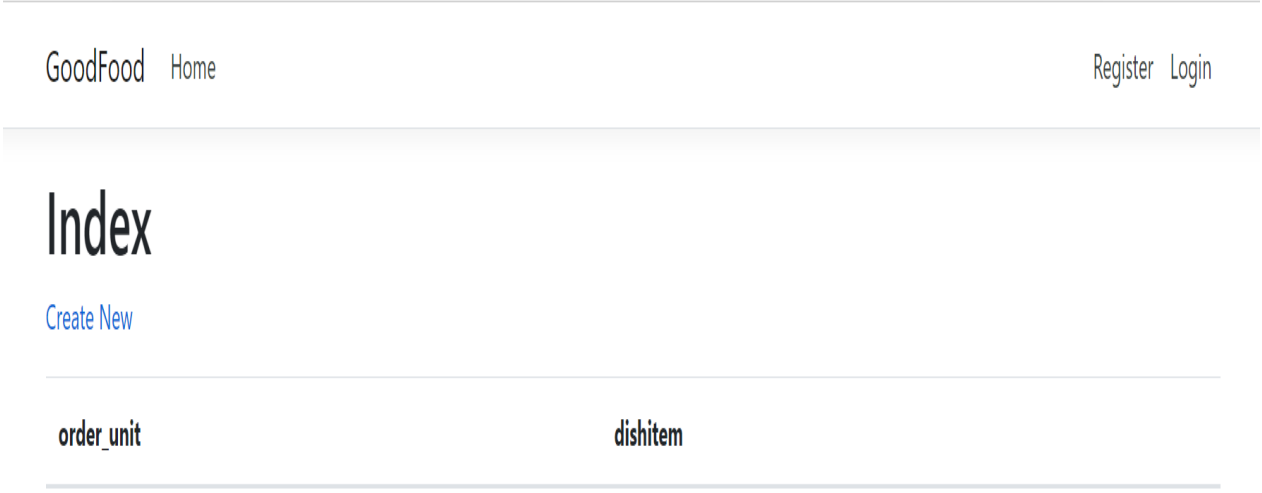


Figure 50:OrderItems deleted

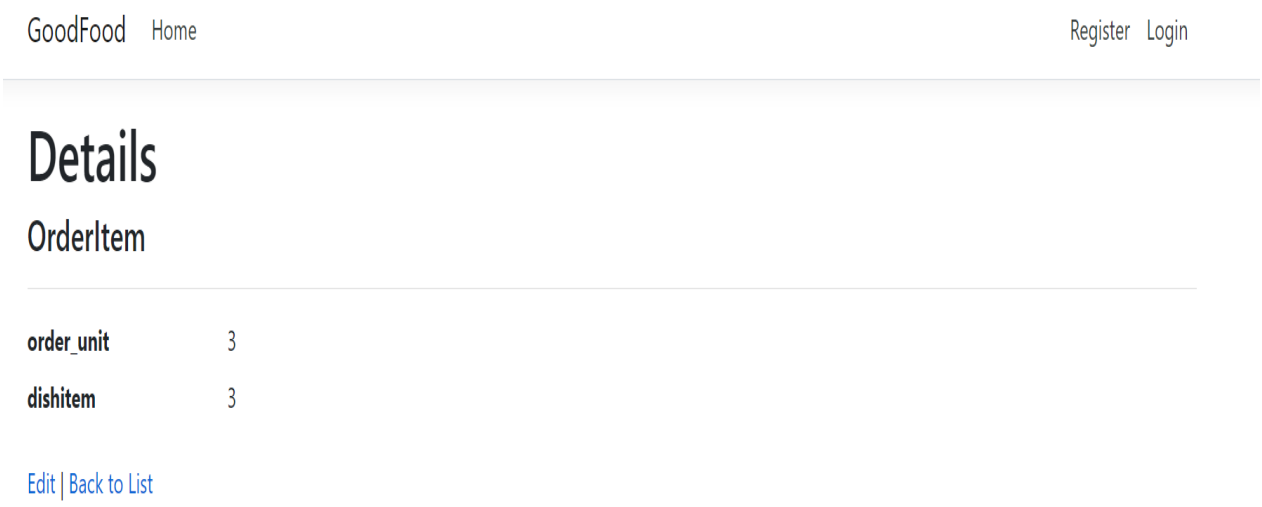


Figure 51:Details of OrderItems

Objective	To test the entire webform named OrderItems.
Expected Result	To insert, edit and delete data.
Outcome	Data's inserted, edited, and deleted.
Result	Data's inserted, edited, and deleted.

7.5. Testing of Dishes:

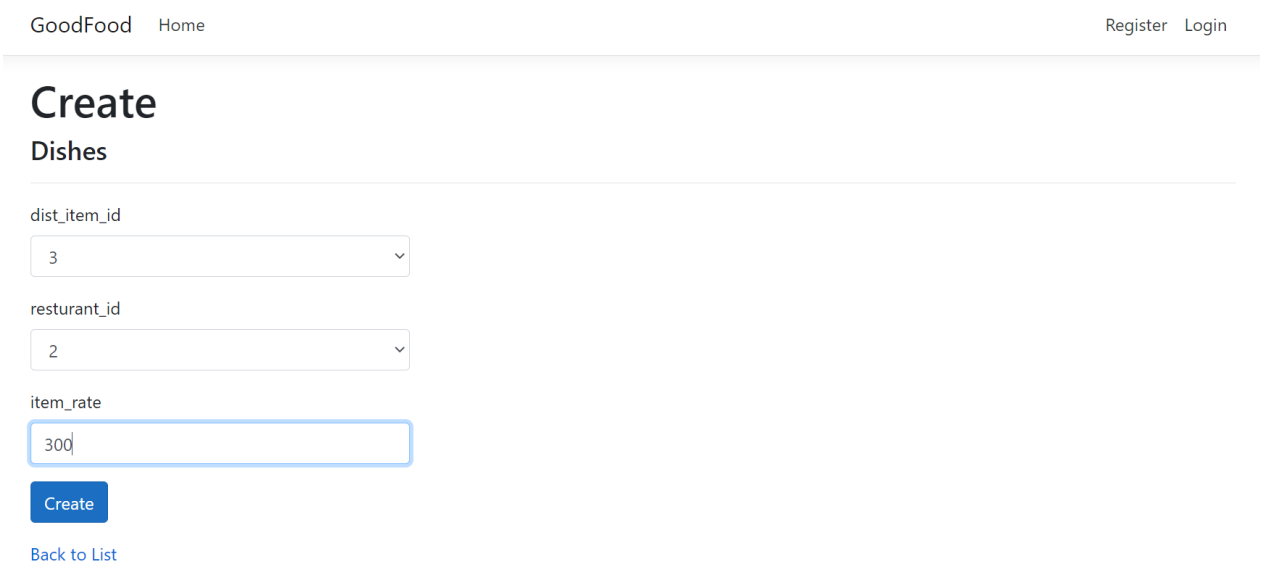


Figure 52:Creating dishes

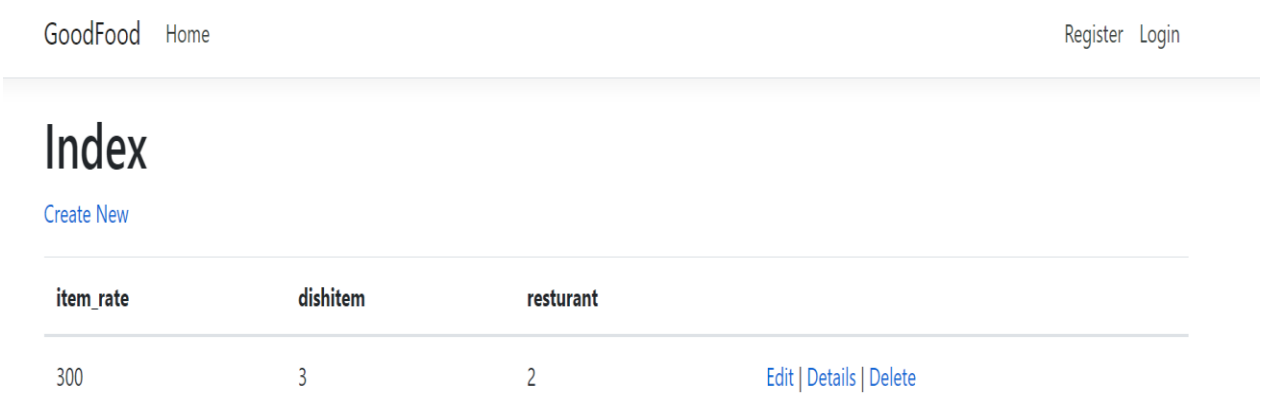


Figure 53:Dishes Created

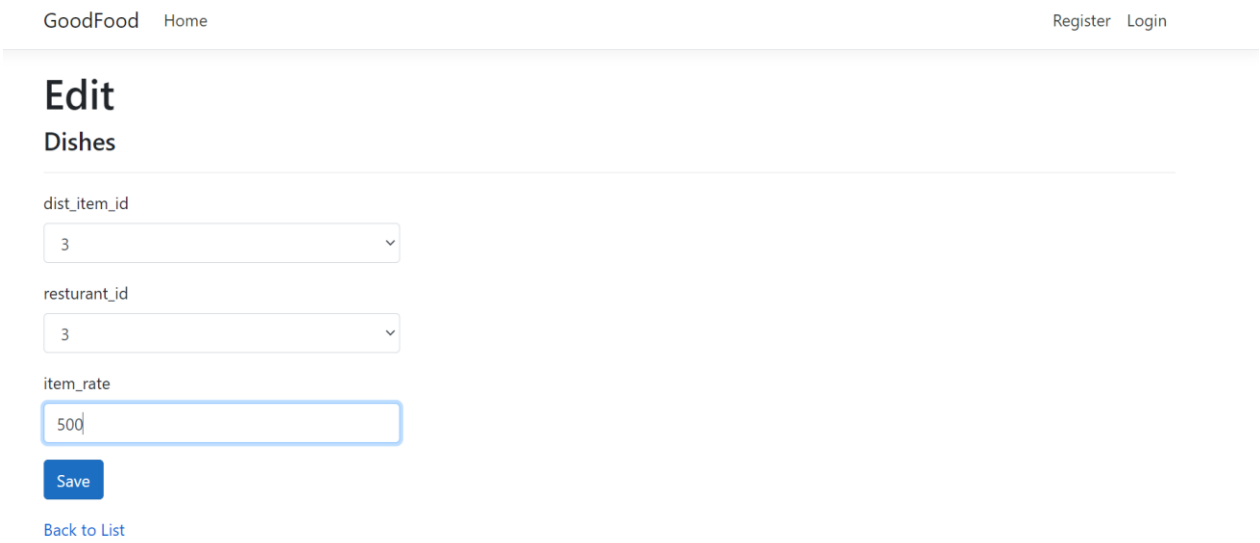


Figure 54:Editing Dishes

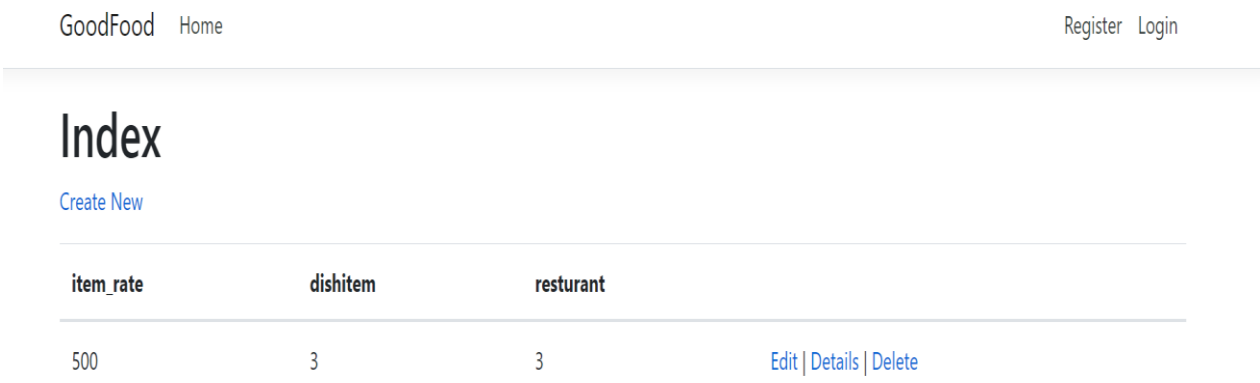


Figure 55:Dishes Edited

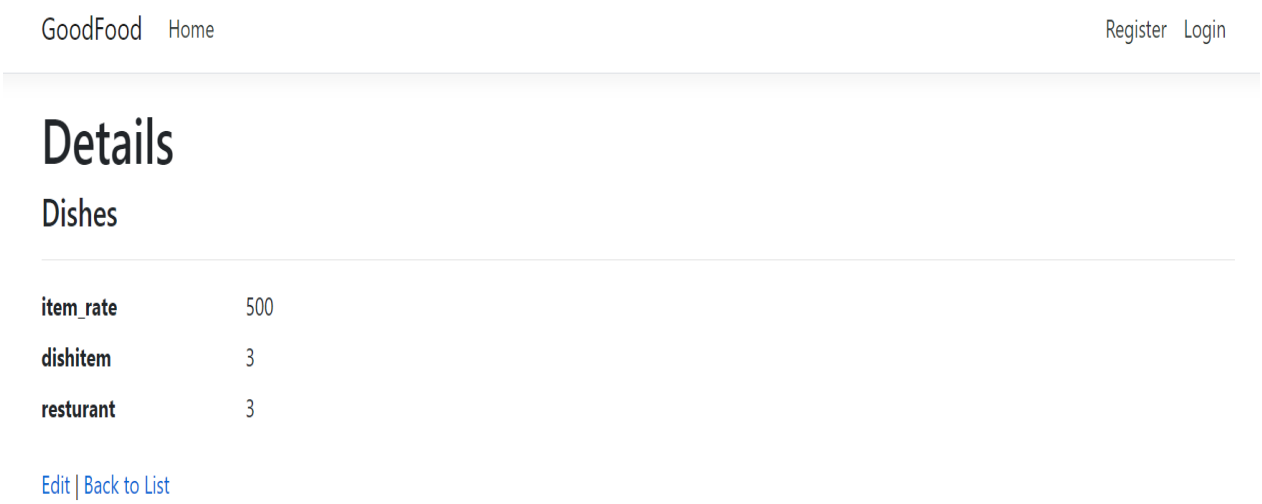


Figure 56:Viewing Details of Dishes

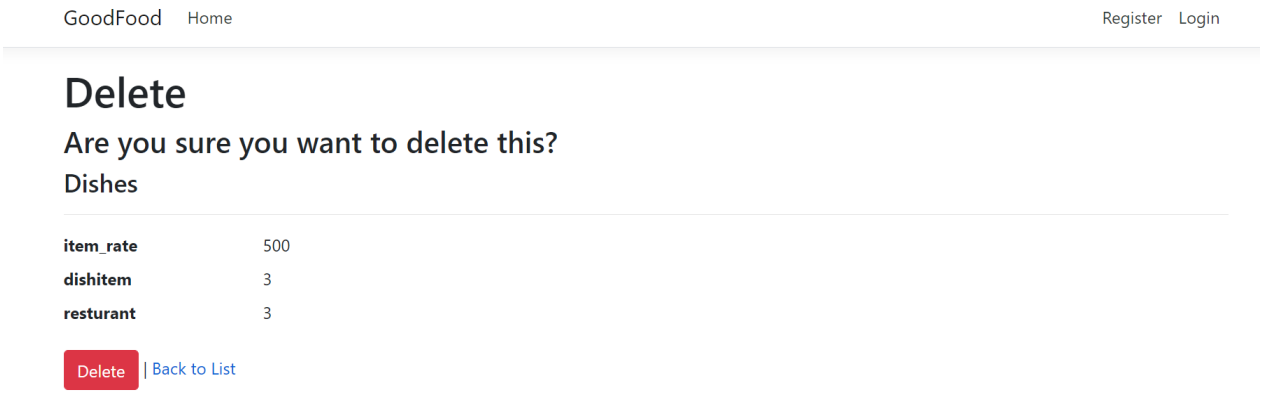


Figure 57:Deleting Dishes

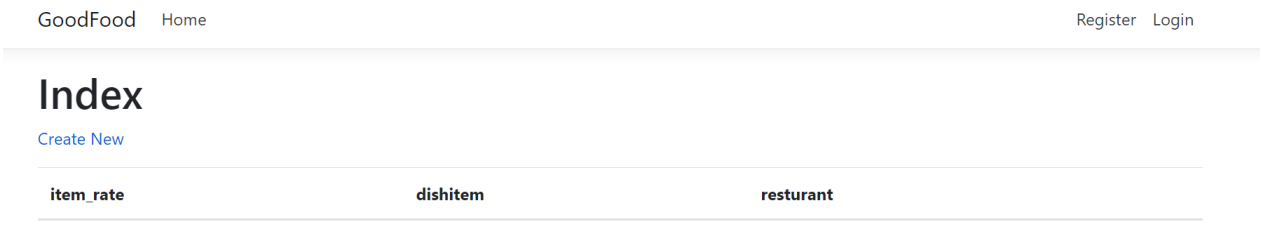


Figure 58:Dishes Deleted

Objective	To test the entire webform named Dishes.
Expected Result	To insert, edit and delete data.
Outcome	Data's inserted, edited, and deleted.
Result	Data's inserted, edited, and deleted.

7.6. Testing of Orders:

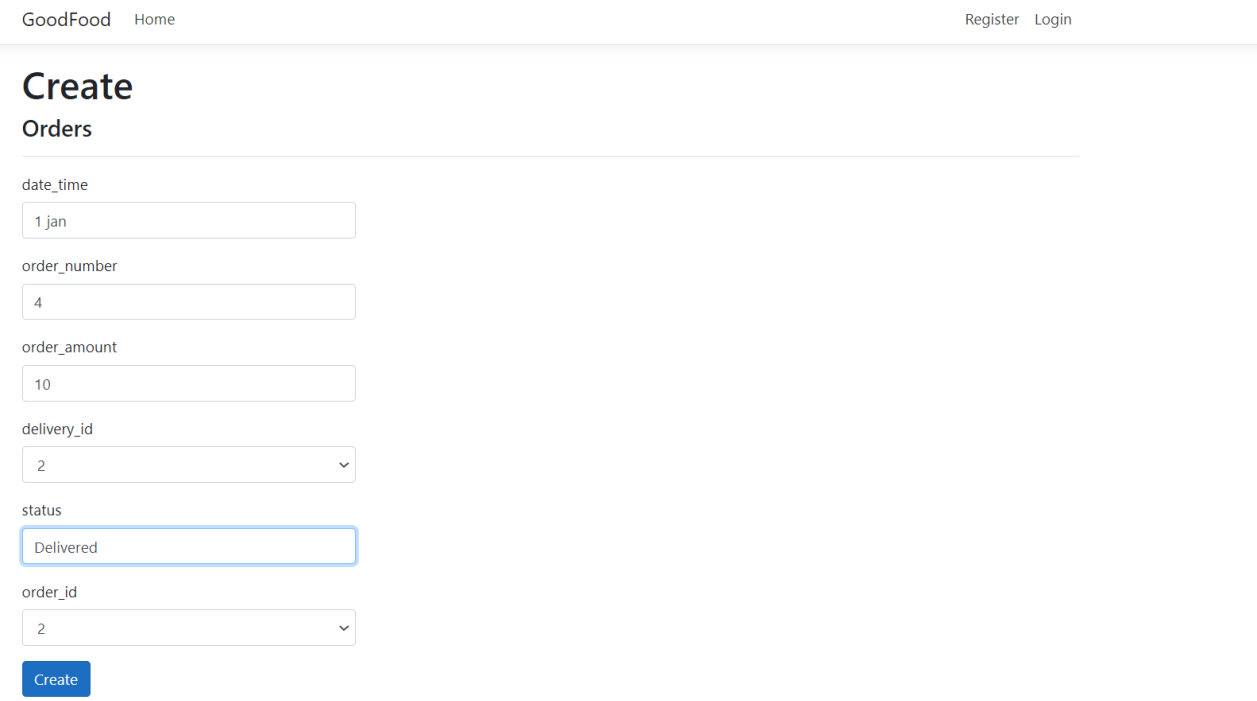


Figure 59:Creating Orders

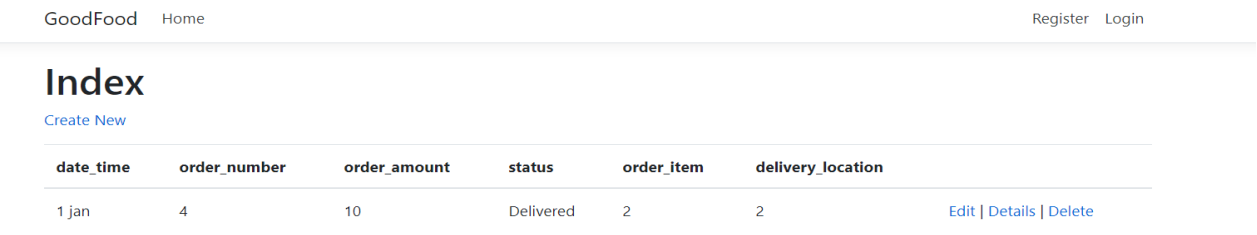


Figure 60:Orders Created

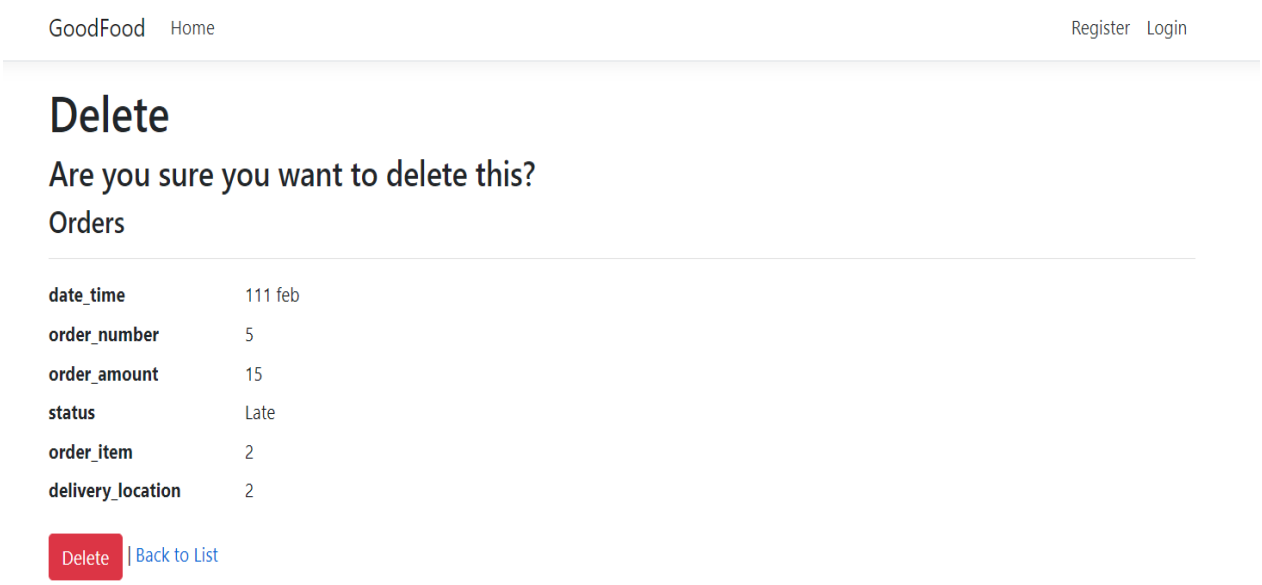


Figure 61:Deleting Orders

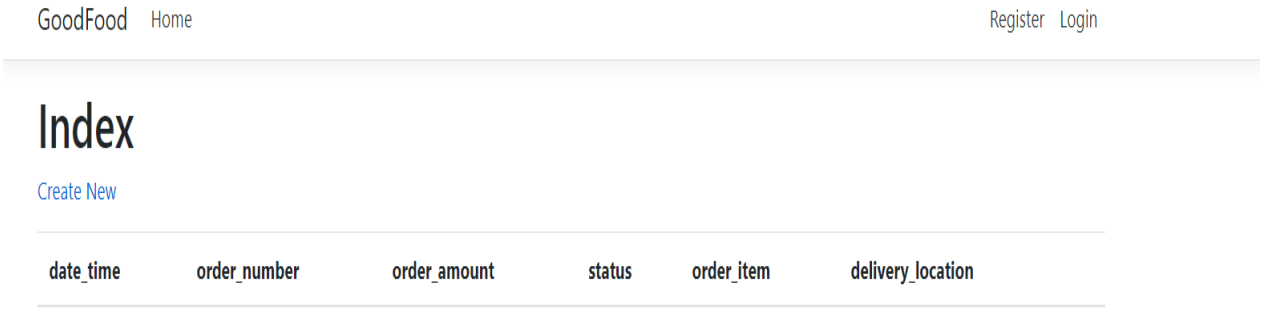


Figure 62:Orders deleted

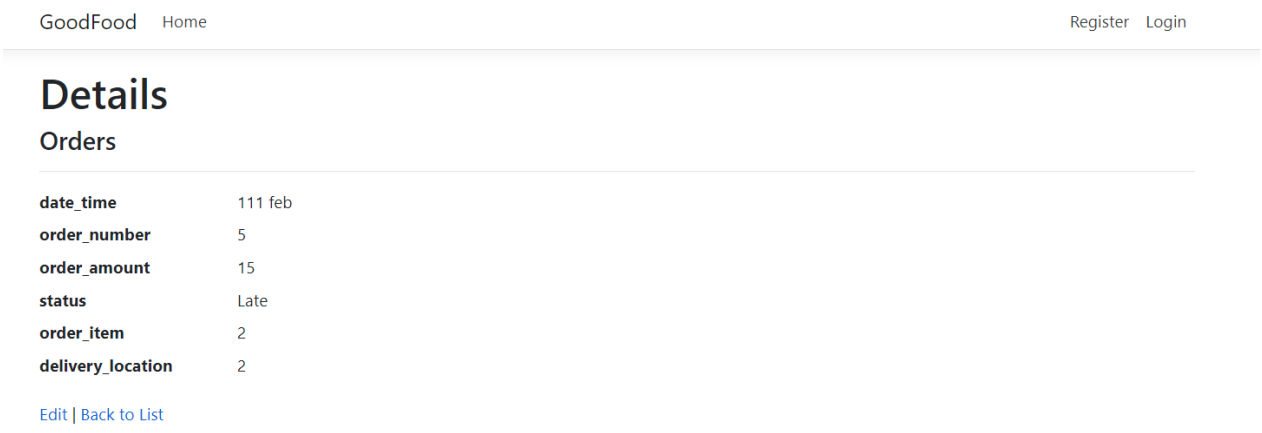


Figure 63:Viewing Details of Orders

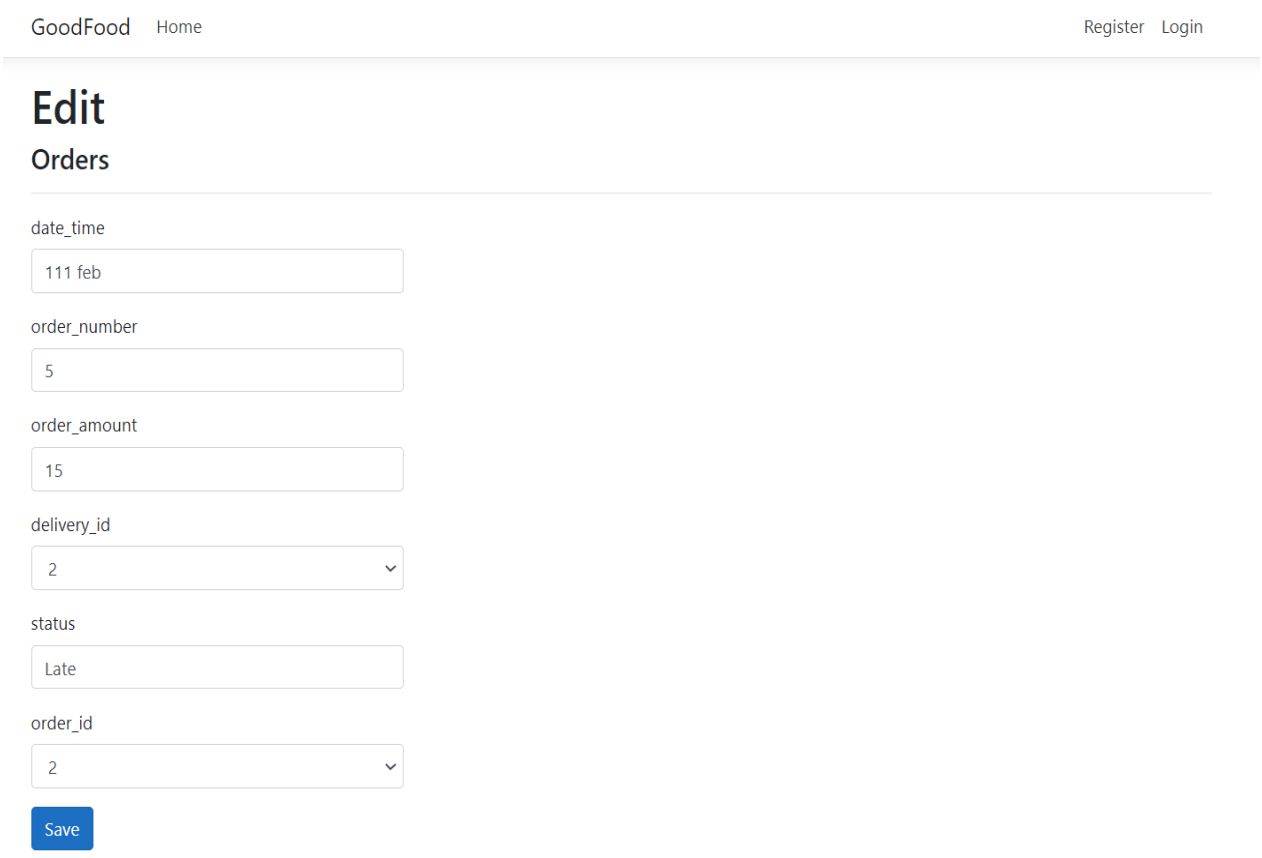


Figure 64:Editing Orders

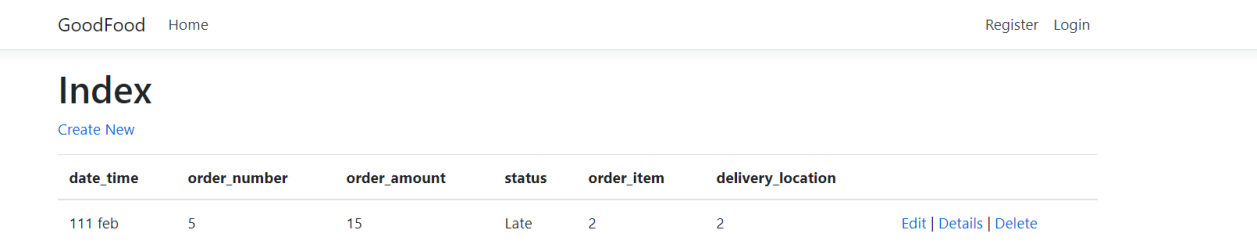
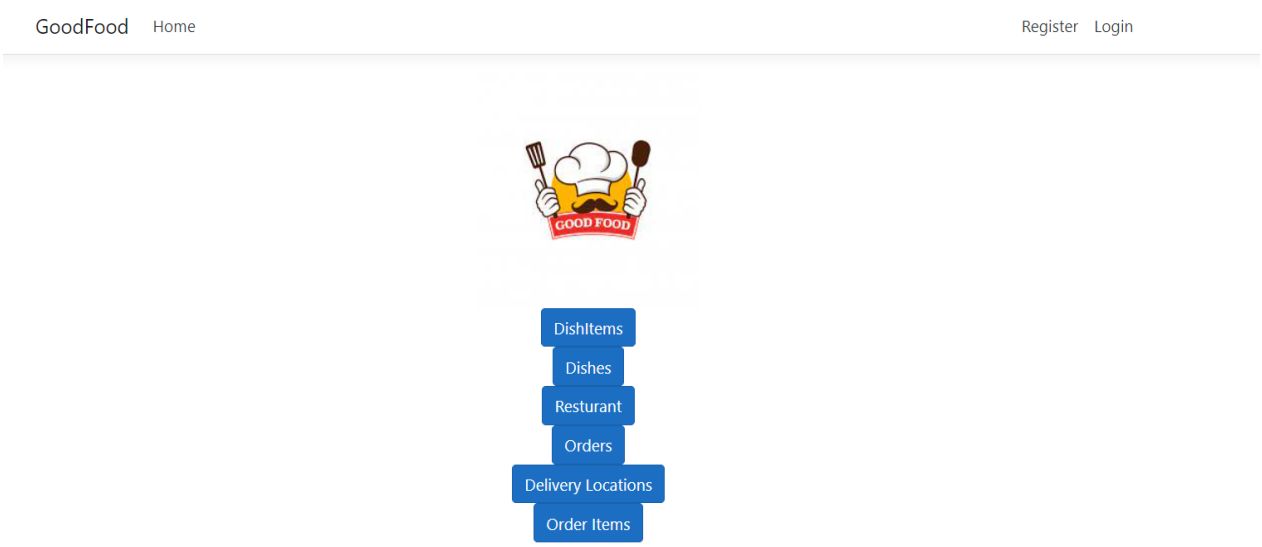


Figure 65:Orders edited

8. User Manual



The above Webform is the GUI (Graphical User Interface) of my project where I have included the appropriate buttons. The following actions are performed in the web form:

i. The DishItems include DishName, DishCode and substitute Name.
ii. The Dishes include itemrate, dishitem and restaurant.
iii. The restaurant includes restaurantName and restaurantcode.
iv. The orders include date_timeorder_numberorder_amountstatusorder_itemdelivery_location.
v. The deliverylocation includes location
vi. The orderItems includes orderunit and ordrItems.

9.FURTHER DISCUSSION:

Through this coursework, a lot of experience has been achieved. The task in the coursework was hard but it was finally completed through lots of research, effort and hard work. At first lots of research were done regarding the different software that were needed to be used. After the research, all the tasks were carried out according to the requirement of the coursework. The coursework did not only complete the tasks given but also helped in developing various skills and taught many things which can be helpful for the career as a database developer. I learned a lot about analyzing the scenario of a company. In brief, an analyzation of working mechanism of travel and tour company. Likewise, I learned to identify the entities and attributes of the system and show the relationship between entities in Oracle SQL Developer Data Modeler. Oracle SQL Developer Modeler was easy to use and made the task a lot easier as it generates the DDL file which contains the generation of table. Similarly, I learned to connect the database from Oracle SQL Developer to ASP.NET with C# (Visual Studio). The Module Teacher Rohit Panday gave different ideas and techniques which made the coursework less hard and its completion in time

Bibliography

https://www.guru99.com/database-normalization.html

https://www.oracle.com/database/technologies/appdev/datamodeler.html
https://stackoverflow.com/questions/6580529/how-to-generate-an-entityrelationship-er-diagram-using-oracle-sql-developer https://www.oracle.com/tools/technologies/whatis-sql-developer.html
https://www.oracle.com/tools/technologies/quickstart-dotnet-for-oracledatabase.html
https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-aspnet-core?view=vs2019
https://www.c-sharpcorner.com/UploadFile/1e050f/insert-update-and-delete-recordin-datagridview-C-Sharp/ https://startbootstrap.com/themes/admin-dashboard/

.