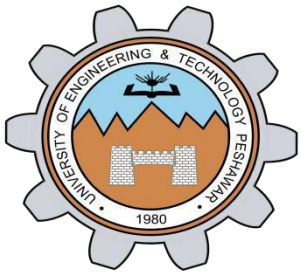
Key Milestone 5: Final Report



DBMS Final Project

Submitted By: Muhammad Saad Amjad Khan

Project Group #19:

Muhammad Ehzaz Khan (22pwcse2108)

Muhammad Saad Amjad Khan (22pwcse2133)

Muhammad Kamil Khan (22pwcse2174)

Section: B

Submitted to: Engr. Sumayyea Salahuddin

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work"

Department of Computer Systems Engineering

University of Engineering and Technology Peshawar

Project: Restaurant Management System

Implementation Details of Restaurant Management System

Introduction

The Restaurant Management System (RMS) is a web-based application developed using Laravel (PHP framework) to streamline and digitize the operations of a restaurant. It handles orders, invoices, menus, customer details, and table bookings efficiently while ensuring a user-friendly interface for both admins and customers.

Technologies Used

- Backend: PHP (Laravel 12)
- Frontend: HTML, CSS, Blade Templates
- Database: MySQL
- Tools: VS Code, XAMPP, Composer
- Libraries: Bootstrap (for UI), Laravel Eloquent ORM

Modules Implemented

User Authentication

- Users can register and log in using Laravel's built-in authentication.
- Admin has privileges to view and manage all modules.
-

Food/Menu Management

- Admin can add, update, and delete food items.
- Each food item includes:
    - Name
    - Description
    - Price
    - Category
    - Image (uploaded and stored in public folder)

Order Management:

- Users can place orders from the available food menu.
- Orders are stored in the database and linked to both user and food entities.
    - Admin can view all orders placed with details such as:
    - Food ordered
    - Quantity
    - Customer information
    - Total cost
    - Order status

Invoice Generation

- Admin can generate invoices for completed orders.
- Each invoice includes:
    - Invoice ID
    - Linked order details
    - Date & time
    - Total bill amount
    - Payment status

Table Reservation

- Users can reserve tables by providing:
    - Name
    - Phone number
    - Date and time
    - Number of people
    - Admin can manage all reservations and mark them as confirmed or cancelled.

Review & Feedback

- Users can submit reviews and feedback after ordering.
- Admin can view and moderate these reviews.

Database Design

Main Tables

- users: Stores user information (id, name, email, password, phone, address).
- foods: Stores menu items (id, name, price, description, image).
- orders: Links users to food items with quantity and status.
- invoices: Generated for completed orders.
- reservations: Stores table booking information.
- reviews: Stores customer reviews for food or service.

Admin Dashboard

- Displays key metrics:
- Total Orders
- Completed Orders
- Revenue Generated
- Reservations Count
- Provides quick links to manage each module.
- Uses Laravel Blade templates with Bootstrap components.

User Interface

- Built using HTML/CSS with Laravel Blade.

- Clean navigation bar.
- Food menu displayed as responsive cards.
- Reservation and order forms with validation.
- Admin pages with tables and actionable buttons.

Error Handling and Validation

- Server-side validation using Laravel's validate() method.
- Try-catch blocks implemented for database operations.
- Custom error pages for 404 and 500.

Security Measures

- Passwords hashed using Laravel's built-in bcrypt().
- Middleware used to protect admin routes.
- CSRF protection enabled by default.
- Mass assignment protection using $fillable.

Testing and Deployment

- Manual testing performed on:
- Order placement
- Invoice generation
- Reservation booking
- Deployed locally using XAMPP.
- Database migrations handled via Laravel Artisan.

Business Rules for Restaurant Management System:

1. users

Business Role:
Stores the details of customers who use the restaurant system. Each user can browse the menu, add items to their cart, place orders, leave reviews, and book tables.

Key Functional Rules:

- One user can have multiple orders.
- One user can post multiple reviews.
- One user can have multiple invoices and carts.
- Each user is uniquely identified by their ID.

2. employees

Business Role:
Manages records of restaurant staff such as waiters, chefs, and managers. Employees are assigned to handle orders and work shifts.

Key Functional Rules:

- Each employee has a designated shift.
- Employees may be linked to handling one or more orders.

3. food

Business Role:
Maintains the master list of food items offered by the restaurant. This includes information like name, description, price, and image for display.

Key Functional Rules:

- Food items appear in the menu.
- Food items are referenced in orders, cart, and reviews.

4. orders

Business Role:
Stores all placed orders made by users. Each order links a user to a food item and the employee responsible for fulfilling it. Contains the order's current status and amount.

Key Functional Rules:

- Each order belongs to one user.
- Each order is assigned to one employee.
- Each order is for one food item (in flattened design).
- Order statuses include: Pending, Completed, Cancelled.

5. tables

Business Role:
Stores information about the physical dining tables in the restaurant including their seating capacity and current availability status.

Key Functional Rules:

- Each table can be reserved (linked via books).
- Table status can be: Available, Reserved, Occupied.

6. books

Business Role:
Manages customer reservations for dining tables. Tracks which user reserved which table at what date and time, for how many guests.

Key Functional Rules:

- A user can make multiple bookings.
- A table can have multiple bookings (on different dates/times).
- Booking includes time and number of guests.

7. reviews

Business Role:
Captures feedback from users about food items. Each review includes a rating and review text, tied to a specific food item.

Key Functional Rules:

- A user can review multiple food items.
- A food item can have many reviews.
- Ratings help in calculating average food quality scores.

8. carts

Business Role:
Temporarily stores food items that a user is planning to order. Acts like a shopping basket before converting to an order.

Key Functional Rules:

- A cart is user-specific.
- Items in the cart are not yet part of any official order.
- Cart is cleared when the order is placed.

9. invoice

Business Role:
Keeps a financial record of each completed order. Used for billing, accounting, and payment tracking.

Key Functional Rules:

- Each invoice belongs to a user and an order.
- Invoices store final amount, status (e.g., Paid, Unpaid), and timestamp.
- One invoice per order.

Main Entities & Their Description:

1. users:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
name	String	name
phone	String	phone
email	String	email
address	String	address
password	string	password

2. employees:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
name	String	name
role	String	role
phone	String	phone
salary	Integer	salary
shift_timing	DateTime	shift
date_of_joining	Date	date_joining

3. foods:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
title	String	title
detail	String	detail

price	Integer	price
image	String	image

4. orders:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
user_id (FK)	Integer	user_id
food_id (FK)	Integer	food_id
Employee_id (FK)	integer	Employee_id
amount	Integer	amount
status	String	status

5. tables:

Attributes	Datatypes	Name in Database
id(PK)	Integer	id
capacity	Integer	capacity
details	String	detail
status	String	status

6. books:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
table_id (FK)	Integer	table_id
phone	String	phone
guests	Integer	guests
date	DateTime	date
time	DateTime	time

7. reviews:

Attributes	Datatypes	Name in Database
id (PK)	Integer	id
food_id (FK)	Integer	food_id
review	String	review
rating	Integer	rating
date	DateTime	date

8. carts:

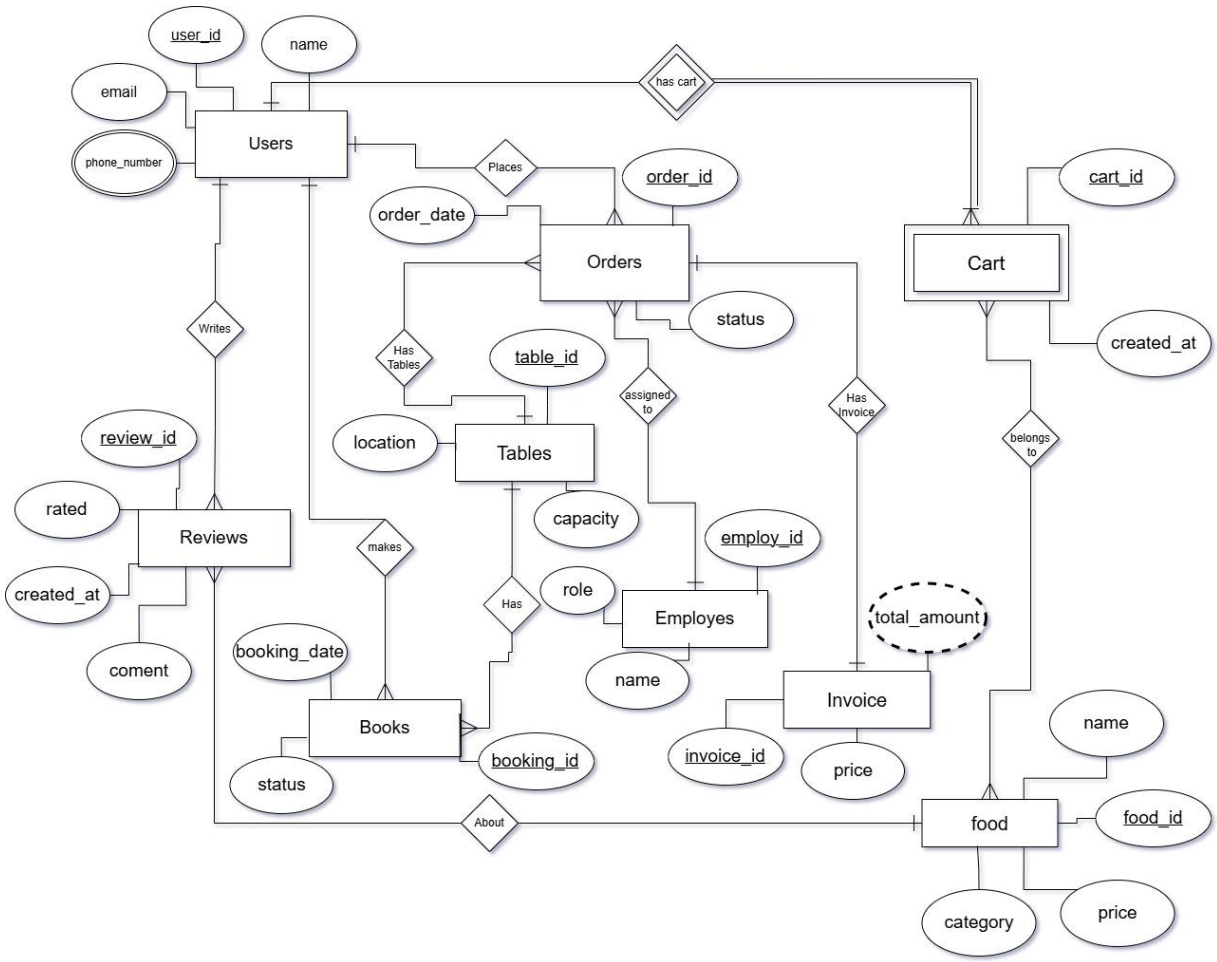
Attributes	Datatypes	Name in Database
id (PK)	Integer	id
user_id (FK)	Integer	user_id
food_id (FK)	Integer	food_id
quantity	Integer	quantity

9. invoice:

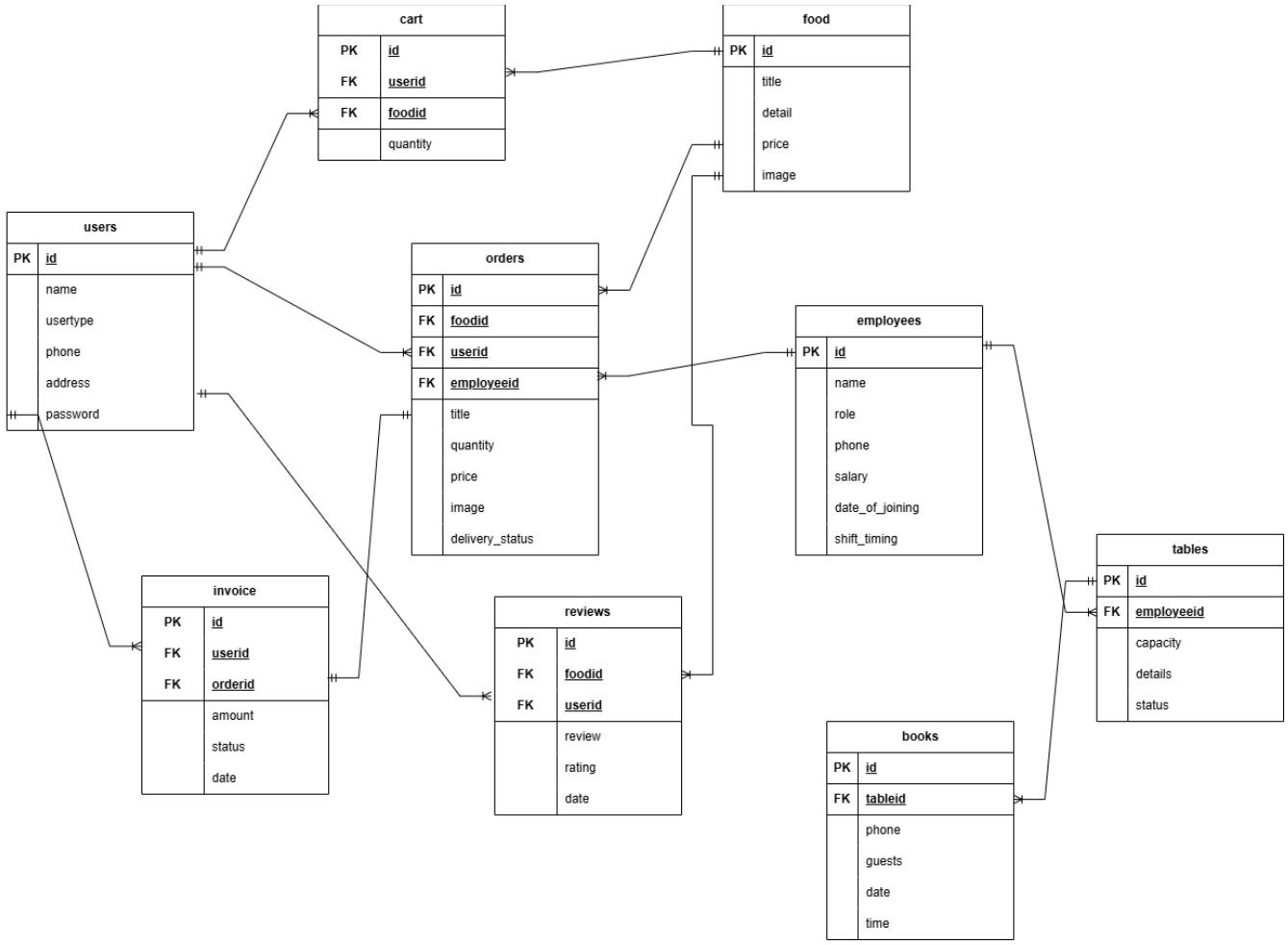
Attributes	Datatypes	Name in Database
id (PK)	Integer	id
order_id (FK)	Integer	order_id
user_id (FK)	Integer	user_id
amount	Integer	amount
status	String	status
date	DateTime	date

Finalized Conceptual Schema:

ERD (Entity Relationship Diagram):



EERD (Enhanced Entity Relationship Diagram):



Final Normalized Forms:

users(id PK, name, phone, email, address, password)

employees(id PK, name, role, phone, salary, date_of_joining, shift_timing)

books(id PK, table_id FK, phone, guests, date)

carts(id PK, user_id FK, food_id FK, quantity)

tables(id PK, details, capacity, status)

orders(id PK, food_id FK, user_id FK, employee_id FK, total_amount, order_status)

reviews(id PK, food_id FK, user_id FK, review, rating, date)

invoice(id PK, order_id FK, user_id FK, amount, status, date)

Table: users

Metadata:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
☐	2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	3	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	4	usertype	varchar(255)	utf8mb4_unicode_ci		No	user			Change Drop More
☐	5	phone	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	6	address	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	7	email_verified_at	timestamp			Yes	NULL			Change Drop More
☐	8	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	9	two_factor_secret	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	10	two_factor_recovery_codes	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	11	two_factor_confirmed_at	timestamp			Yes	NULL			Change Drop More
☐	12	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	13	current_team_id	bigint(20)		UNSIGNED	Yes	NULL			Change Drop More
☐	14	profile_photo_path	varchar(2048)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
☐	15	created_at	timestamp			Yes	NULL			Change Drop More
☐	16	updated_at	timestamp			Yes	NULL			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Console

Sample Data:

←T→	id	name	email	usertype	phone	address	email_verified_at	password
☐ Edit Copy Delete	1	User	user@gmail.com	user	12345678	asd	NULL	$2y$12$cMkJrdxSvUmaooLnlweHtOfjtJsAIya7eJxnCDYw1yl...
☐ Edit Copy Delete	2	Admin	admin@gmail.com	admin	12345678	Abbotabad	NULL	$2y$12$whGpCq1ez5.qCKPlsP2i7eSMbhSl2QDPK0osaERm2z4...
☐ Edit Copy Delete	3	Saad Amjad	saad@gmail.com	user	03079966105	UET Peshawar	NULL	$2y$12$JAZBgkGHr5GG5SaS4gDYsuCkhnGhUhBufcW1CYhjt8d...
☐ Edit Copy Delete	4	Ehzaz	ehzaz@gmail.com	user	03250749754	UET Peshawar	NULL	$2y$12$PsIlkC7AVhmwsNWOb7jc9.VpsYQvmaJkSWnFLKXJSjV...
☐ Edit Copy Delete	5	Employee1	employee1@gmail.com	employee	12345	asd	NULL	$2y$12$hwwJ7wPIOHuzgy7ZT1KeU.2ma2mVm3sssCRJRWlkVkw..

SQL Queries:

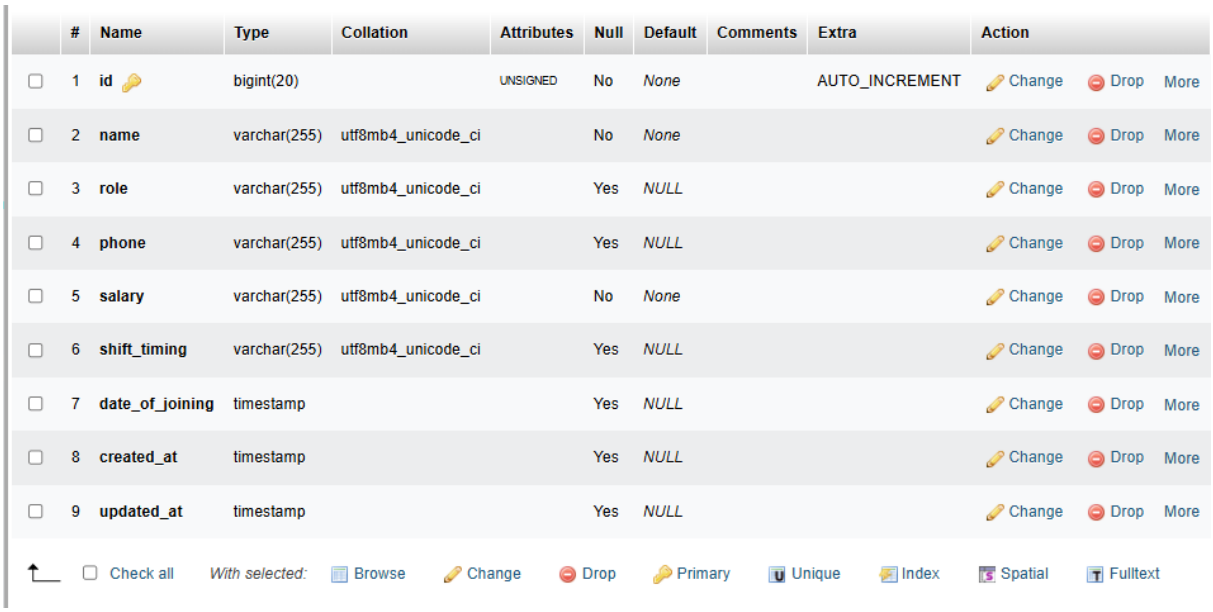
```
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    phone VARCHAR(15),
    email VARCHAR(100),
    address TEXT,
    password VARCHAR(100)
);
```

```
INSERT INTO users (id, name, email, usertype, phone, address, password)

VALUES (2, 'Admin', 'admin@gmail.com', 'admin', '12345678', 'Abbotabad', '12345678');
```

```
INSERT INTO users (id, name, email, usertype, phone, address, password)

VALUES (3, 'Saad Amjad', 'saad@gmail.com', 'user', '03079966105', 'UET Peshawar',
'12345678');
```

Table: employees

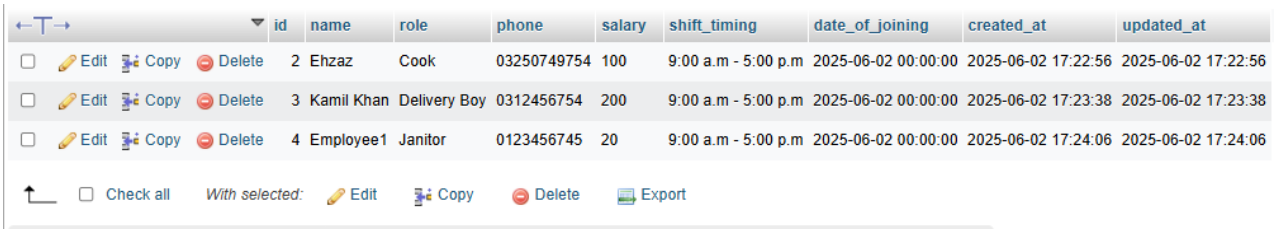
Metadata:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id 🔑	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	name	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	3	role	varchar(255)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	4	phone	varchar(255)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	5	salary	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	6	shift_timing	varchar(255)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
☐	7	date_of_joining	timestamp			Yes	*NULL*			Change Drop More
☐	8	created_at	timestamp			Yes	*NULL*			Change Drop More
☐	9	updated_at	timestamp			Yes	*NULL*			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Sample Data:



	id	name	role	phone	salary	shift_timing	date_of_joining	created_at	updated_at
Edit Copy Delete	2	Ehzaz	Cook	03250749754	100	9:00 a.m - 5:00 p.m	2025-06-02 00:00:00	2025-06-02 17:22:56	2025-06-02 17:22:56
Edit Copy Delete	3	Kamil Khan	Delivery Boy	0312456754	200	9:00 a.m - 5:00 p.m	2025-06-02 00:00:00	2025-06-02 17:23:38	2025-06-02 17:23:38
Edit Copy Delete	4	Employee1	Janitor	0123456745	20	9:00 a.m - 5:00 p.m	2025-06-02 00:00:00	2025-06-02 17:24:06	2025-06-02 17:24:06

Check all With selected: Edit Copy Delete Export

SQL Queries:

```
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    role VARCHAR(50),
    phone VARCHAR(15),
    salary INT,
    shift TIMESTAMP,
    date_joining DATE
);
```

```
INSERT INTO employees (id, name, role, phone, salary, shift_timing, date_of_joining)
VALUES (2, 'Ehzaz', 'Cook', '03250749754', 100, '9:00 a.m - 5:00 p.m', '2025-06-02 00:00:00');
```

```
INSERT INTO employees (id, name, role, phone, salary, shift_timing, date_of_joining)
VALUES (3, 'Kamil Khan', 'Delivery Boy', '0312456754', 200, '9:00 a.m - 5:00 p.m', '2025-06-02 00:00:00);
```

```
INSERT INTO employees (id, name, role, phone, salary, shift_timing, date_of_joining)
VALUES (4, 'Employee1', 'Janitor', '0123456745', 20, '9:00 a.m - 5:00 p.m', '2025-06-02 00:00:00');
```

Table: food

Metadata:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
3	detail	longtext	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
4	price	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	image	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	created_at	timestamp			Yes	NULL			Change Drop More
7	updated_at	timestamp			Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Sample Data:

	id	title	detail	price	image	created_at	updated_at
Edit Copy Delete	4	Pizza	Chicken Tikka with topping enjoy with 69 Sauces	1500	1747506463.jpg	2025-05-17 18:27:43	2025-05-25 19:10:26
Edit Copy Delete	5	Burger	Beef Burger with special Mayo	800	1747588652.jpg	2025-05-18 17:17:32	2025-05-18 17:17:32
Edit Copy Delete	7	Pasta	Special Pasta	1800	1747588870.jpg	2025-05-18 17:21:10	2025-05-18 17:21:10
Edit Copy Delete	8	Soup	Korean Soup	2000	1747592316.jpg	2025-05-18 18:18:36	2025-05-18 18:18:36

Check all With selected: Edit Copy Delete Export

SQL Queries:

```
CREATE TABLE food_item (
   id INT PRIMARY KEY,
   title VARCHAR(100),
   detail TEXT,
   price INT,
   image VARCHAR(255)
);
```

```
INSERT INTO food (id, title, detail, price, image)
VALUES (4, 'Pizza', 'Chicken Tikka with topping enjoy with 69 Sauces', 1500, '1747506463.jpg');
```

INSERT INTO food (id, title, detail, price, image)

VALUES (5, 'Burger', 'Beef Burger with Special Mayo', 800, ‘1747588652.jpg’);

INSERT INTO food (id, title, detail, price, image)

VALUES (7, 'Pasta', 'Special Pasta', 1800, ‘1747588870.jpg’);

INSERT INTO food (id, title, detail, price, image)

VALUES (8, 'Soup', Korean Soup', 2000, ‘1747592316.jpg’);

Table: orders

Metadata:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
☐	2	user_id	bigint(20)		UNSIGNED	No	None			Change Drop More
☐	3	food_id	bigint(20)		UNSIGNED	No	None			Change Drop More
☐	4	employee_id	bigint(20)		UNSIGNED	Yes	NULL			Change Drop More
☐	5	amount	int(11)			No	None			Change Drop More
☐	6	status	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	7	created_at	timestamp			Yes	NULL			Change Drop More
☐	8	updated_at	timestamp			Yes	NULL			Change Drop More

Sample Data:

←T→	id	user_id	food_id	employee_id	amount	status	created_at	updated_at
☐ Edit Copy Delete	10	1	1	1	80000	Pending	2025-06-20 11:33:19	2025-06-20 11:33:19
☐ Edit Copy Delete	11	1	1	1	97600	Pending	2025-06-20 11:34:52	2025-06-20 11:34:52
☐ Edit Copy Delete	12	4	1	1	24000	Pending	2025-06-20 13:34:16	2025-06-20 13:34:16

SQL Queries:

CREATE TABLE orders (
  id INT PRIMARY KEY,

```
    user_id INT,
    food_id INT,
    employee_id INT,
    amount INT,
    status VARCHAR(50),
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (food_id) REFERENCES food_item(id),
    FOREIGN KEY (employee_id) REFERENCES employees(id)
);

INSERT INTO orders (id, user_id, food_id, employee_id, amount, status)
VALUES (10, 1, 1, 1, 8000, Pending)

INSERT INTO orders (id, user_id, food_id, employee_id, amount, status)
VALUES (11, 1, 1, 1, 8000, Pending)

INSERT INTO orders (id, user_id, food_id, employee_id, amount, status)
VALUES (12, 1, 1, 1, 8000, Pending)
```

Table: tables

Metadata:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	capacity	int(11)			No	None			Change Drop More
3	details	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
4	status	varchar(255)	utf8mb4_unicode_ci		No	available			Change Drop More
5	created_at	timestamp			Yes	NULL			Change Drop More
6	updated_at	timestamp			Yes	NULL			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Sample Data:

	id	capacity	details	status	created_at	updated_at
Edit Copy Delete	5	4	Close to window	Availible	2025-06-02 11:14:21	2025-06-02 15:28:48
Edit Copy Delete	6	3	In the middle of the restaurent	Available	2025-06-02 17:35:41	2025-06-02 17:35:41
Edit Copy Delete	7	6	Near the window that ponits towards road.	Occupied	2025-06-02 17:36:19	2025-06-02 17:36:19
Edit Copy Delete	8	2	On Balcony	Available	2025-06-02 17:36:47	2025-06-02 17:36:47

SQL Queries:

```
CREATE TABLE tables (
    id INT PRIMARY KEY,
    capacity INT,
    detail TEXT,
    status VARCHAR(50)
);
```

INSERT INTO tables (id, capacity, details, status)

VALUES (5, 4, ‘Close to window’, ‘Availible’)

INSERT INTO tables (id, capacity, details, status)

VALUES (6, 3, ‘In the middle of the restaurant’, ‘Available’)

INSERT INTO table (id, capacity, details, status)

VALUES (7, 6, ‘Near the window that points towards road.’, ‘Occupied’)

INSERT INTO tables (id, capacity, details, status)

VALUES (8, 2, ‘On Balcony’, ‘Available’)

Table: reviews

Metadata:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	user_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	3	food_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	4	review	text	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	5	rating	int(11)			No	*None*			Change Drop More
☐	6	date	datetime			No	*None*			Change Drop More

Sample Data:

	id	user_id	food_id	review	rating	date
Edit Copy Delete	1	1	1	Fantastic	4	2025-06-20 09:59:33
Edit Copy Delete	2	1	1	Not Good	2	2025-06-20 10:00:10

SQL Queries:

```
CREATE TABLE reviews (
   id INT PRIMARY KEY,
   user_id INT,
   food_id INT,
   review TEXT,
   rating INT,
   date DATETIME,
   FOREIGN KEY (food_id) REFERENCES food_item(id)
);
```

INSERT INTO reviews (id, user_id, food_id, review, rating, date)

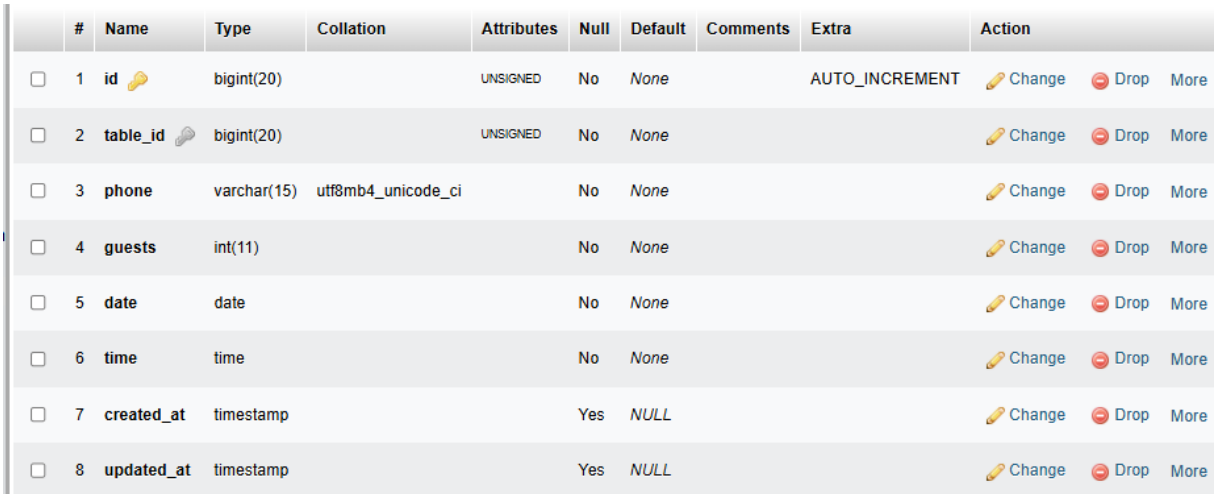
VALUES (1, 1, 1, ‘Fantastic’, 4, 2025-06-20)

INSERT INTO reviews (id, user_id, food_id, review, rating, date)

VALUES (2, 1, 1, ‘Not Good’, 2, 2025-06-20)

Table: books

Metadata:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	table_id	bigint(20)		UNSIGNED	No	None			Change Drop More
3	phone	varchar(15)	utf8mb4_unicode_ci		No	None			Change Drop More
4	guests	int(11)			No	None			Change Drop More
5	date	date			No	None			Change Drop More
6	time	time			No	None			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

Sample Data:

	id	table_id	phone	guests	date	time	created_at	updated_at
Edit Copy Delete	1	1	12345	4	2025-06-26	19:23:00	2025-06-20 10:25:17	2025-06-20 10:25:17

SQL Queries:

```
CREATE TABLE books (
   id INT PRIMARY KEY,
   table_id INT,
   phone VARCHAR(15),
   guests INT,
   date DATETIME,
   time DATETIME,
   FOREIGN KEY (table_id) REFERENCES tables(id)
);
```

```
INSERT INTO books (id, table_id, phone, guest, date, time)

VALUES (1, 1, +923429233995, 4, ‘2025-06-26’, ’22:44’)
```

Table: carts

Metadata:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	user_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	3	food_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	4	quantity	int(11)			No	*None*			Change Drop More
☐	5	price	decimal(8,2)			No	*None*			Change Drop More
☐	6	created_at	timestamp			Yes	*NULL*			Change Drop More
☐	7	updated_at	timestamp			Yes	*NULL*			Change Drop More

Sample Data:

←T→	id	user_id	food_id	quantity	price	created_at	updated_at
☐ Edit Copy Delete	9	4	1	23	18400.00	2025-06-20 13:45:18	2025-06-20 13:45:18
☐ Edit Copy Delete	10	4	1	1	800.00	2025-06-20 13:46:04	2025-06-20 13:46:04

SQL Queries:

```
CREATE TABLE carts (
    id INT PRIMARY KEY,
    user_id INT,
    food_id INT,
    quantity INT,
     price INT,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (food_id) REFERENCES food_item(id)
);

INSERT INTO carts (id, user_id, food_id, quantity, price)
VALUES (9, 4, 1, 23, 18400)
```

INSERT INTO carts (id, user_id, food_id, quantity, price)
VALUES (10, 4, 1, 1, 800)

Table: invoice

Metadata:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	order_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	3	user_id	bigint(20)		UNSIGNED	No	*None*			Change Drop More
☐	4	amount	int(11)			No	*None*			Change Drop More
☐	5	status	varchar(50)	utf8mb4_unicode_ci		No	Pending			Change Drop More
☐	6	date	datetime			No	*None*			Change Drop More
☐	7	created_at	timestamp			Yes	*NULL*			Change Drop More
☐	8	updated_at	timestamp			Yes	*NULL*			Change Drop More

Sample Data:

←T→	id	order_id	user_id	amount	status	date	created_at	updated_at
☐ Edit Copy Delete	4	10	1	80000	Cash on Delivery	2025-06-20 11:33:19	2025-06-20 11:33:19	2025-06-20 11:33:19
☐ Edit Copy Delete	5	11	1	97600	Cash on Delivery	2025-06-20 11:34:52	2025-06-20 11:34:52	2025-06-20 11:34:52
☐ Edit Copy Delete	6	12	4	24000	Cash on Delivery	2025-06-20 13:34:16	2025-06-20 13:34:16	2025-06-20 13:34:16

SQL Queries:

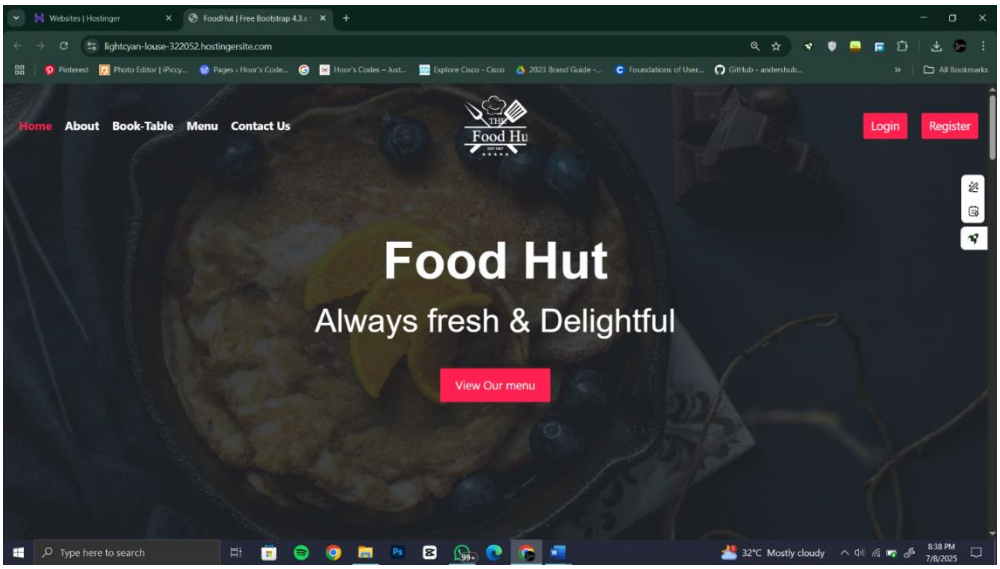
```
CREATE TABLE invoice (
   id INT PRIMARY KEY,
   order_id INT,
   user_id INT,
   amount INT,
   status VARCHAR(50),
   date DATETIME,
   FOREIGN KEY (order_id) REFERENCES orders(id),
   FOREIGN KEY (user_id) REFERENCES users(id));
```

```
INSERT INTO invoice (id, order_id, user_id, amount, status, date)
VALUES (4, 10, 1, 80000, 'Cash on Delivery', 2025-06-20)
```

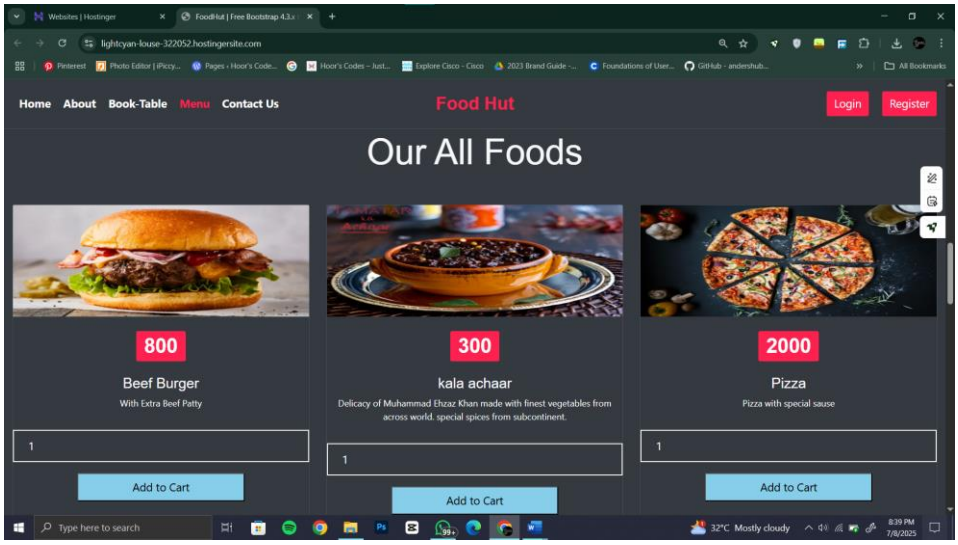
INSERT INTO invoice (id, order_id, user_id, amount, status, date)
VALUES (5, 11, 1, 97600, ‘Cash on Delivery’, 2025-06-20)

COMPREHENSIVE IMPLEMENTATION DETAILS (LARAVEL):

Home page:



Food menu:



Login page:

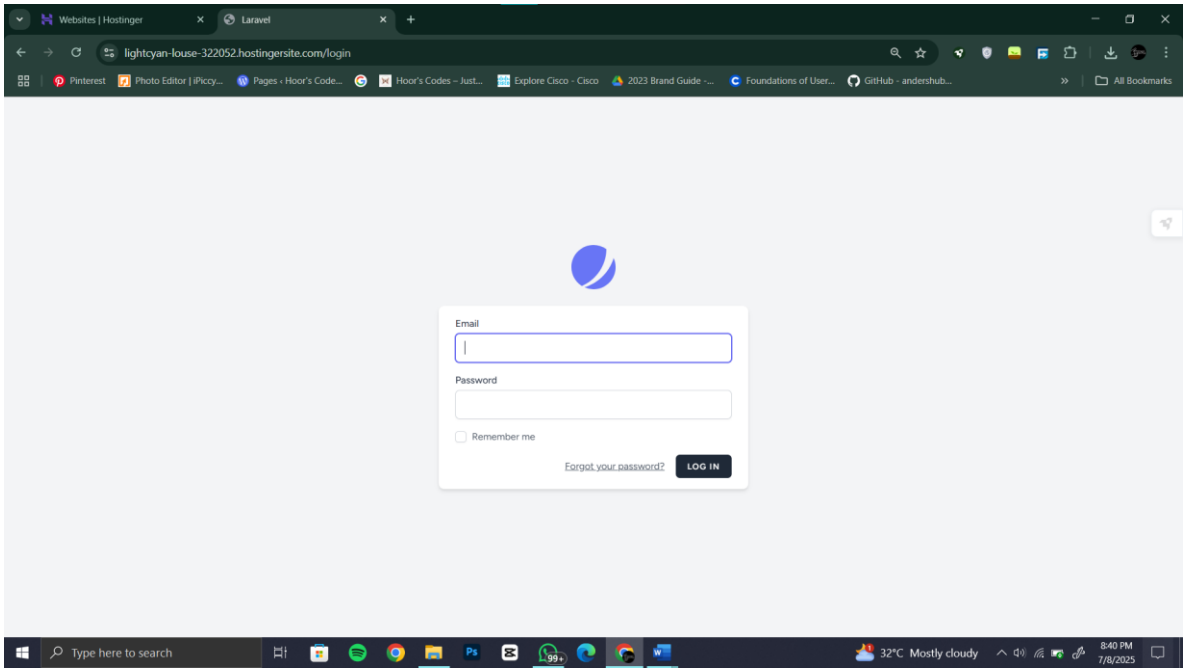
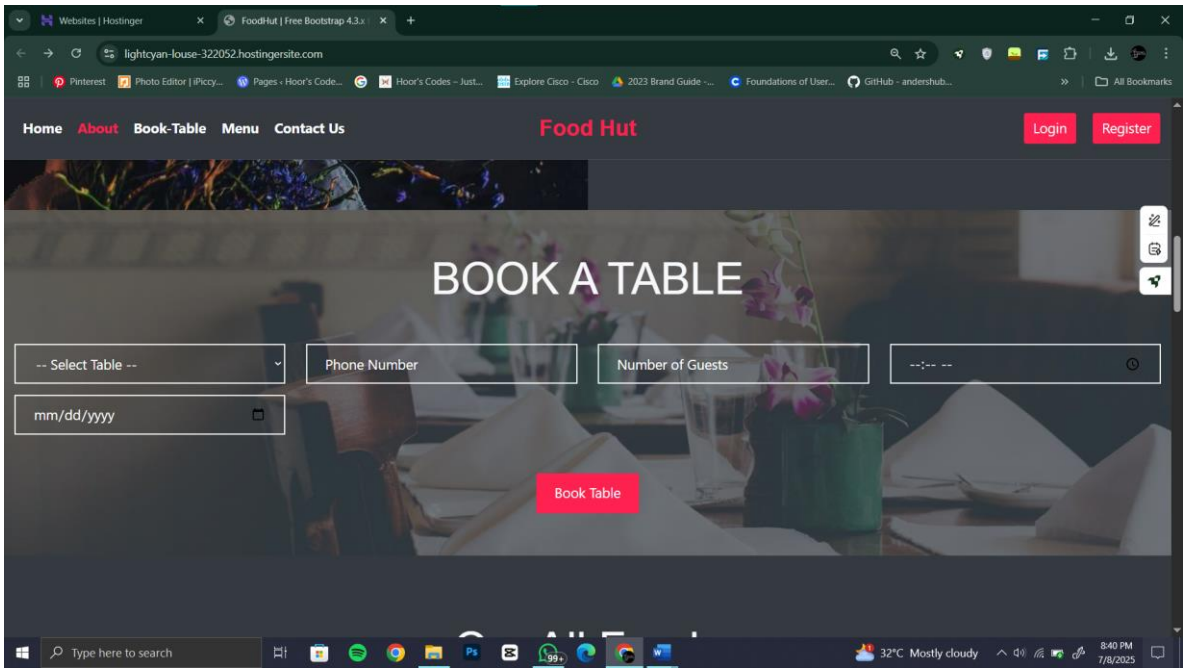
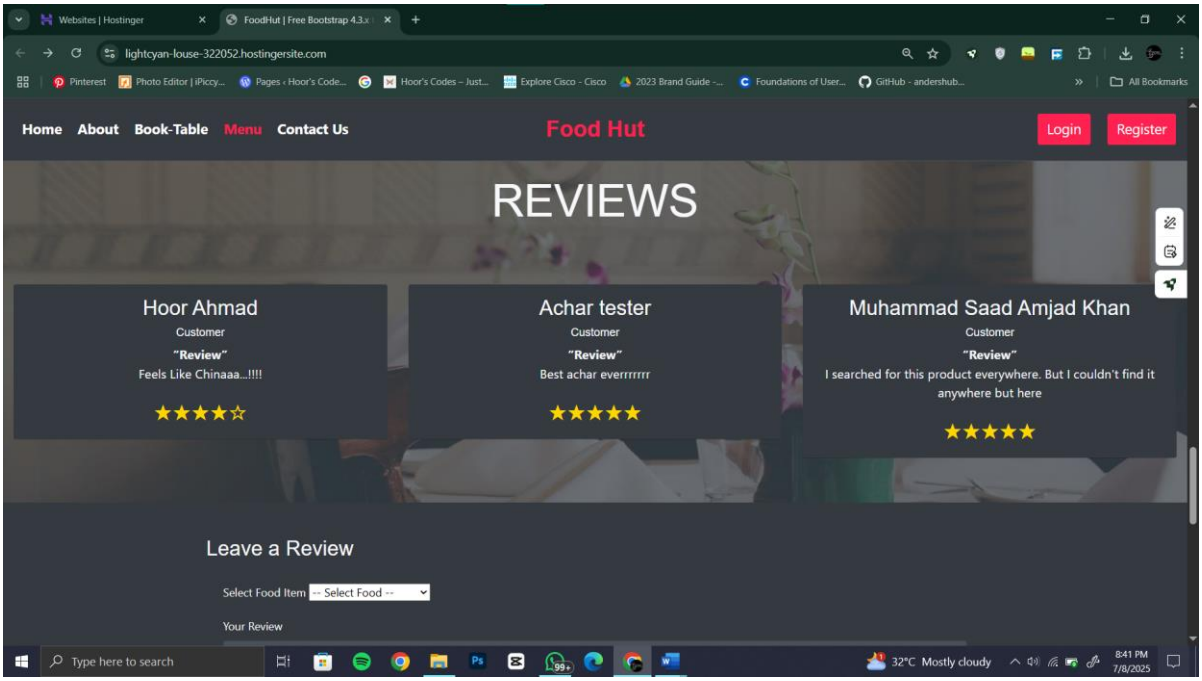


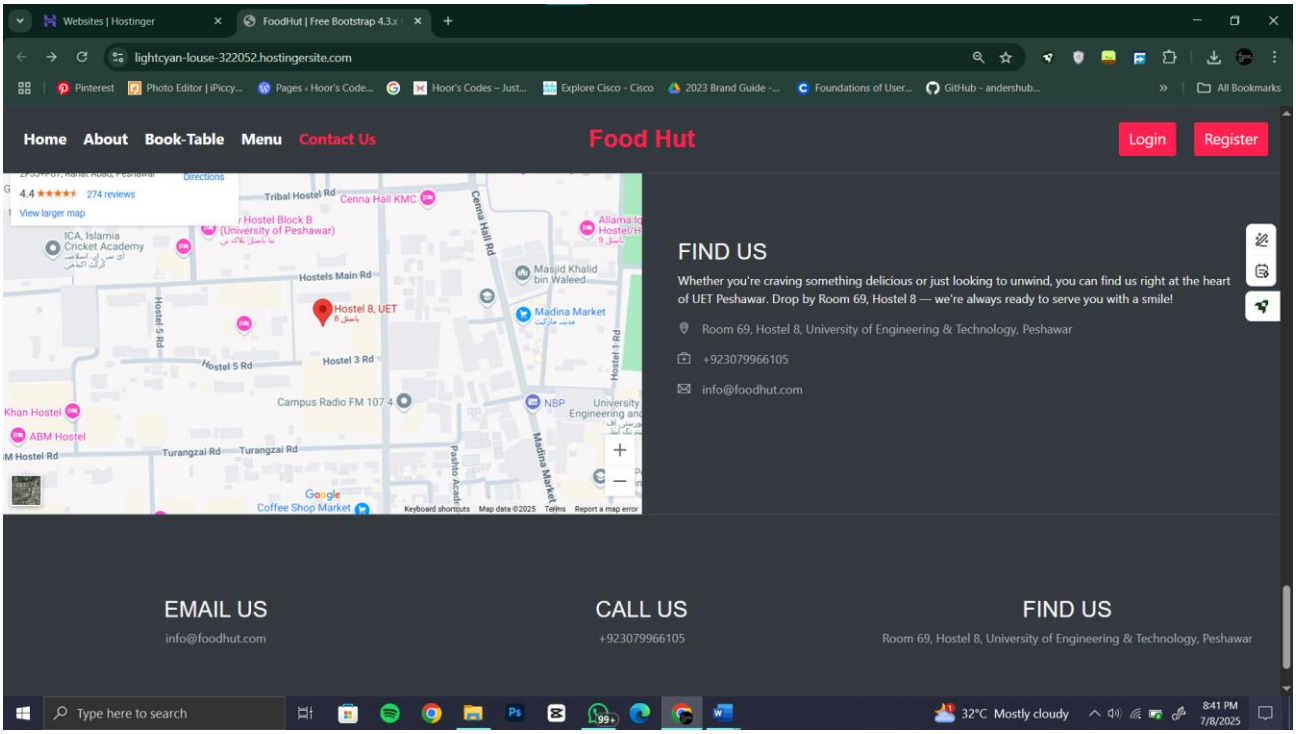
Table booking:



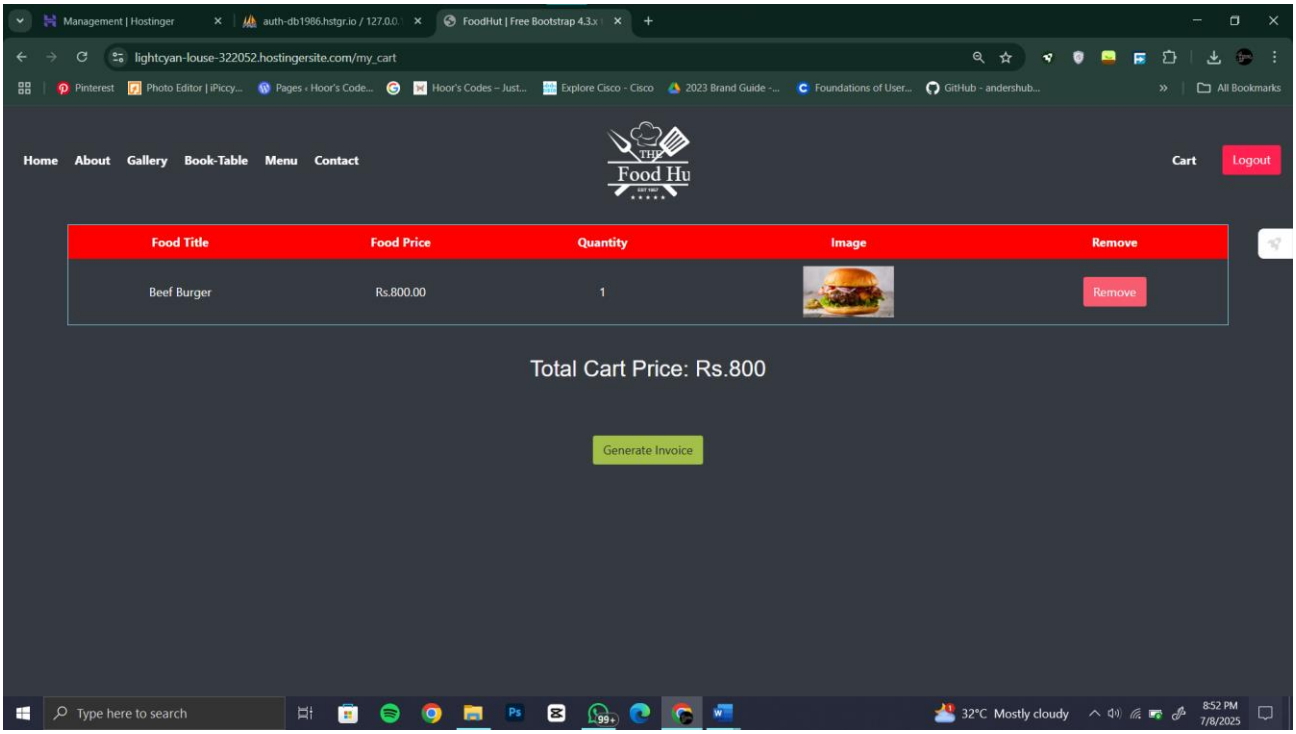
Reviews :



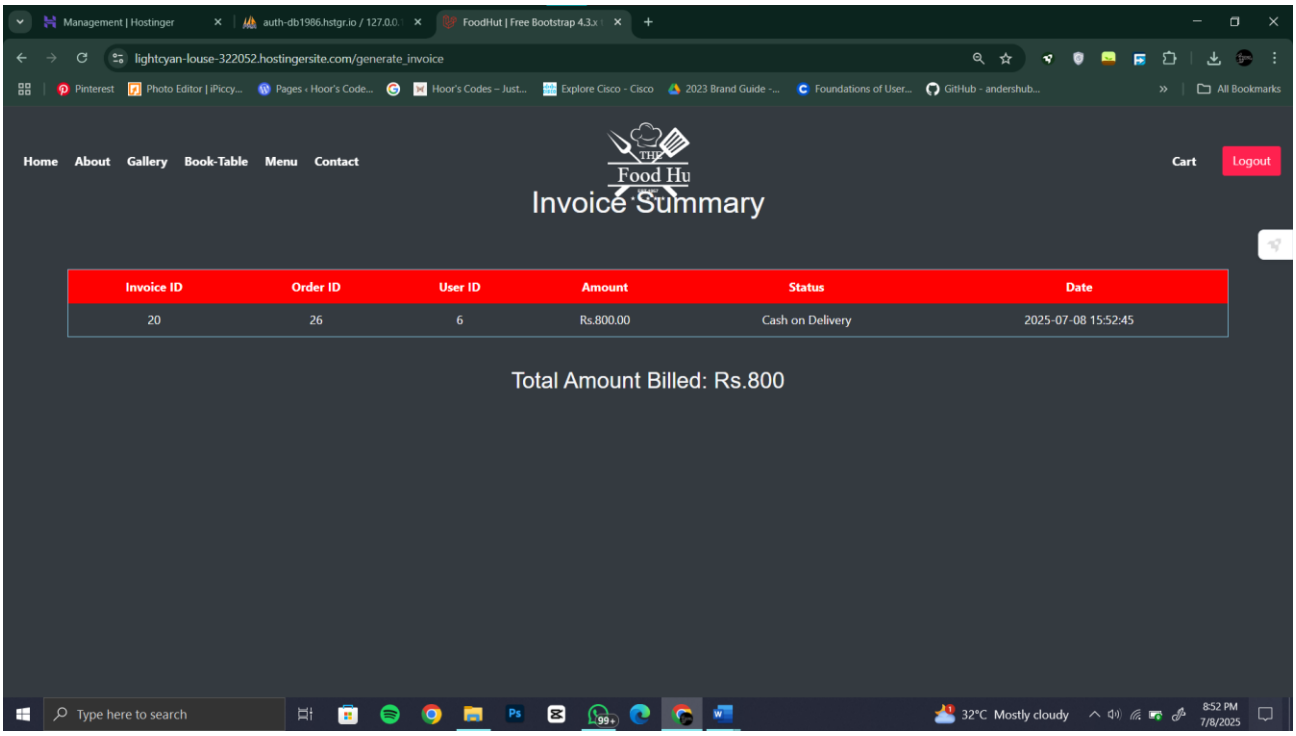
Contact us:



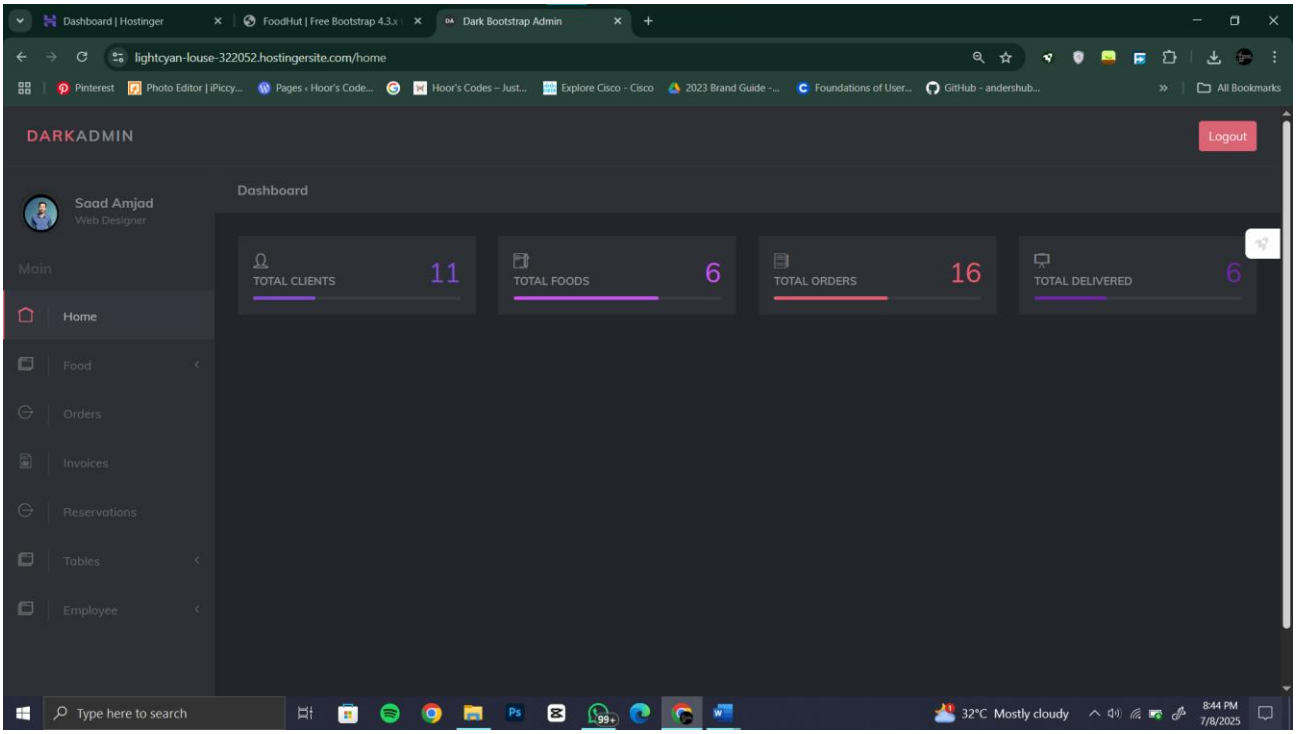
View cart from customer credentials:



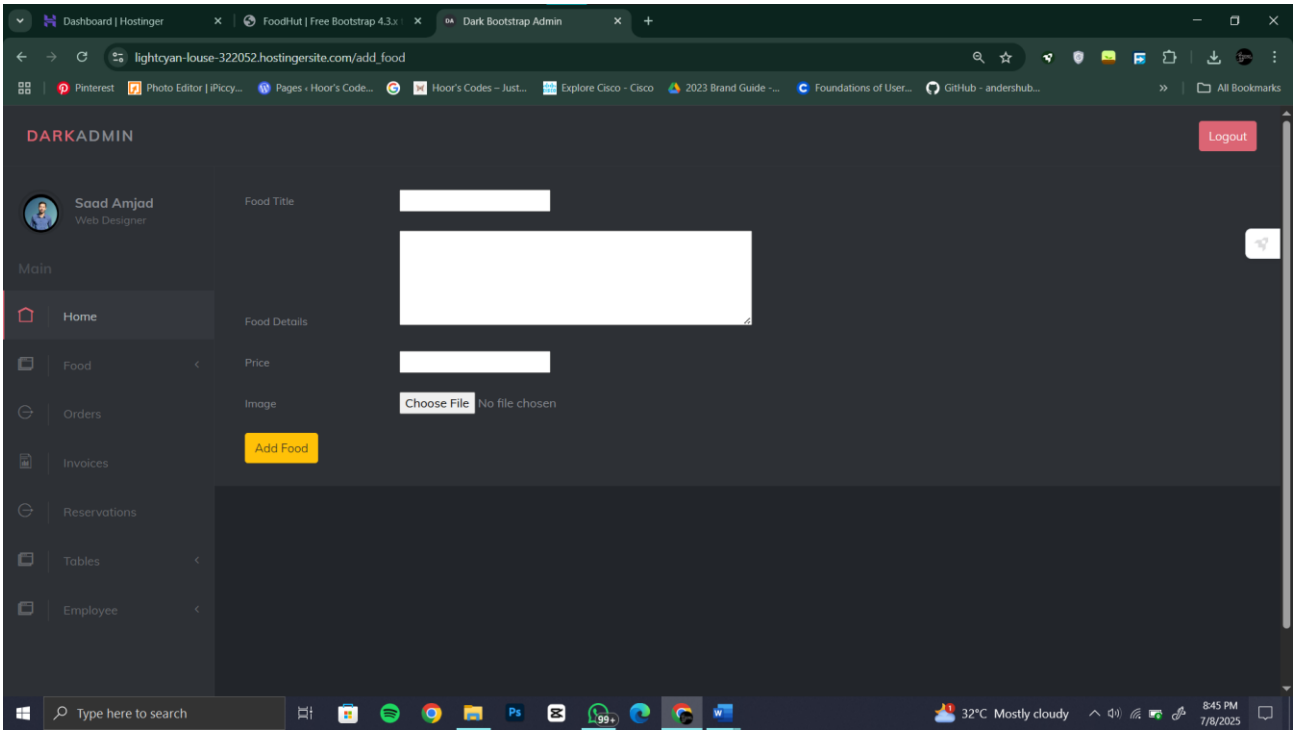
Generate invoice for customer:



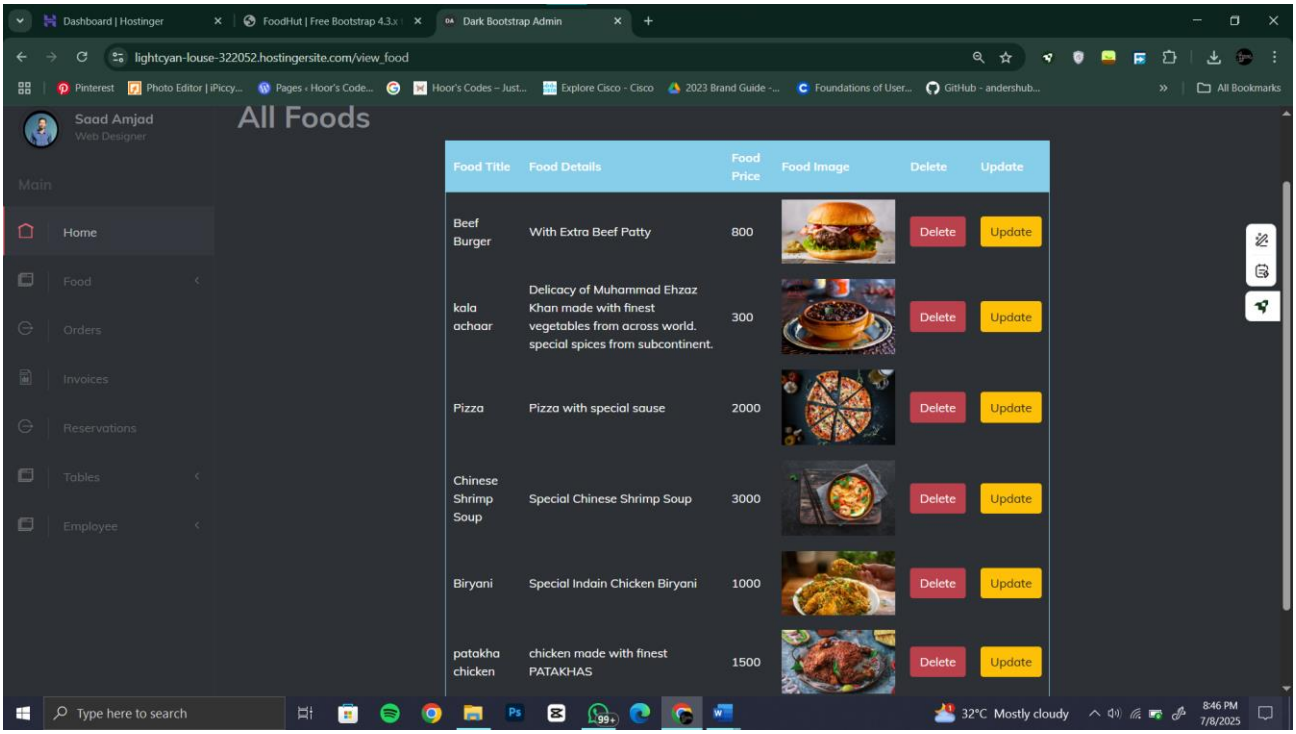
Admin dashboard:



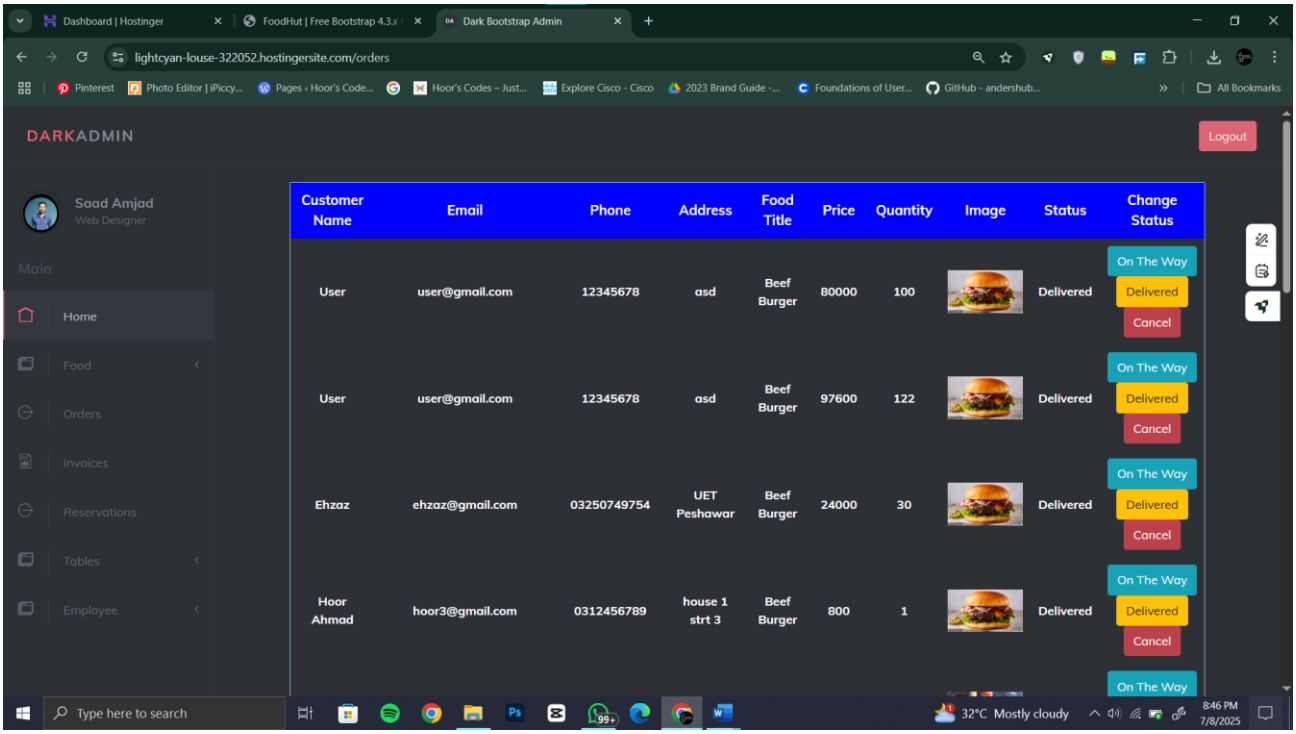
Add food from admin panel:



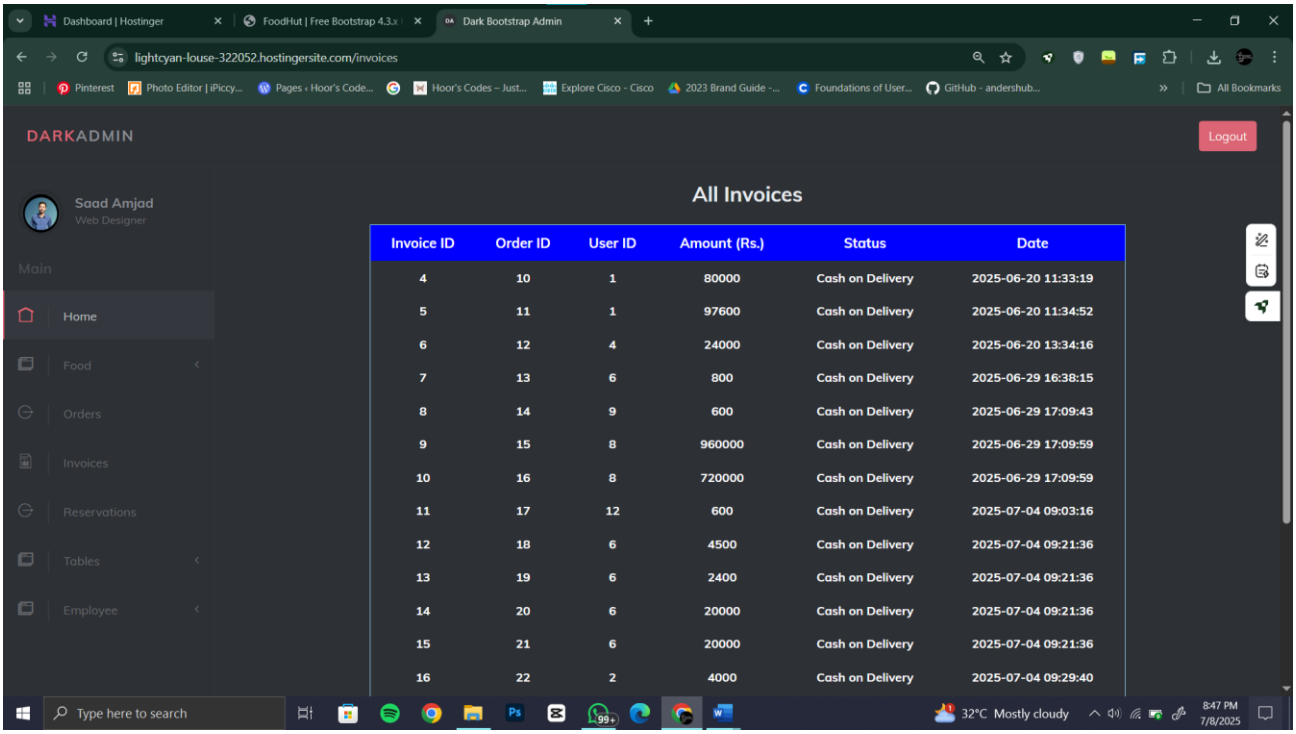
View all the food from admin panel:



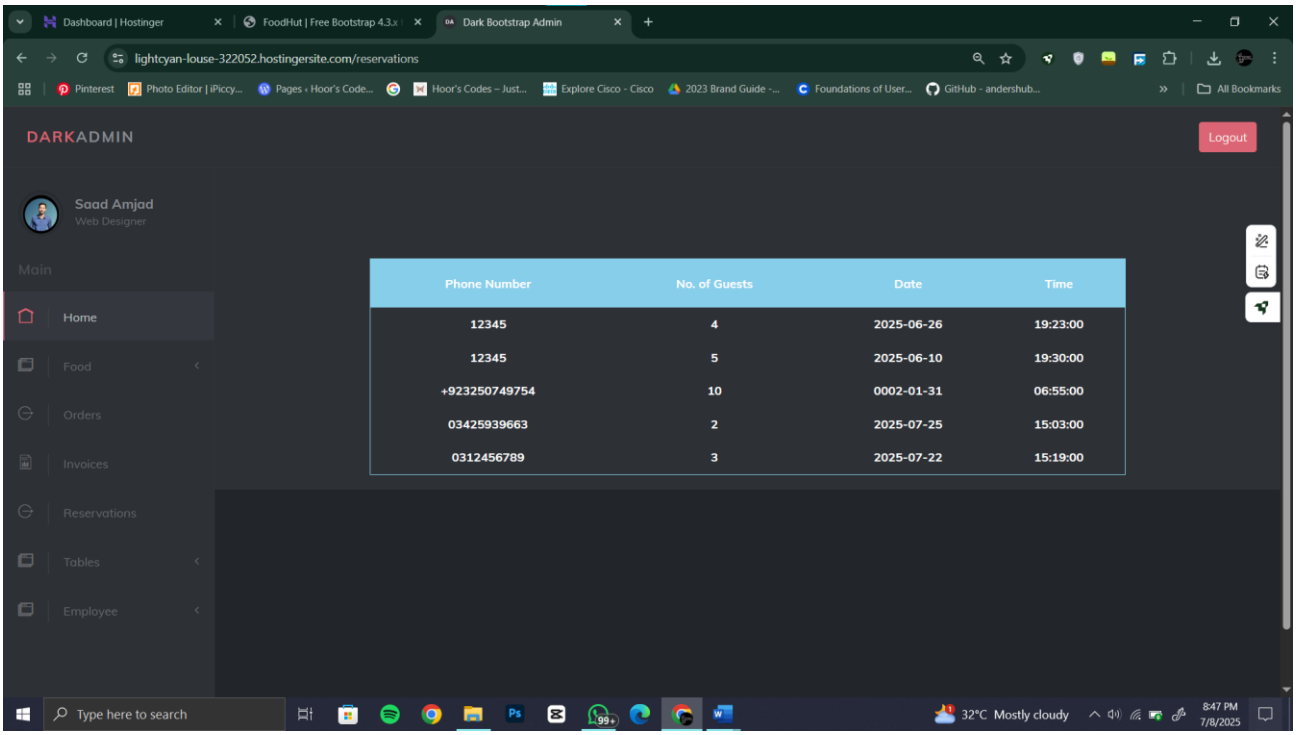
View all the orders and their status from admin panel:



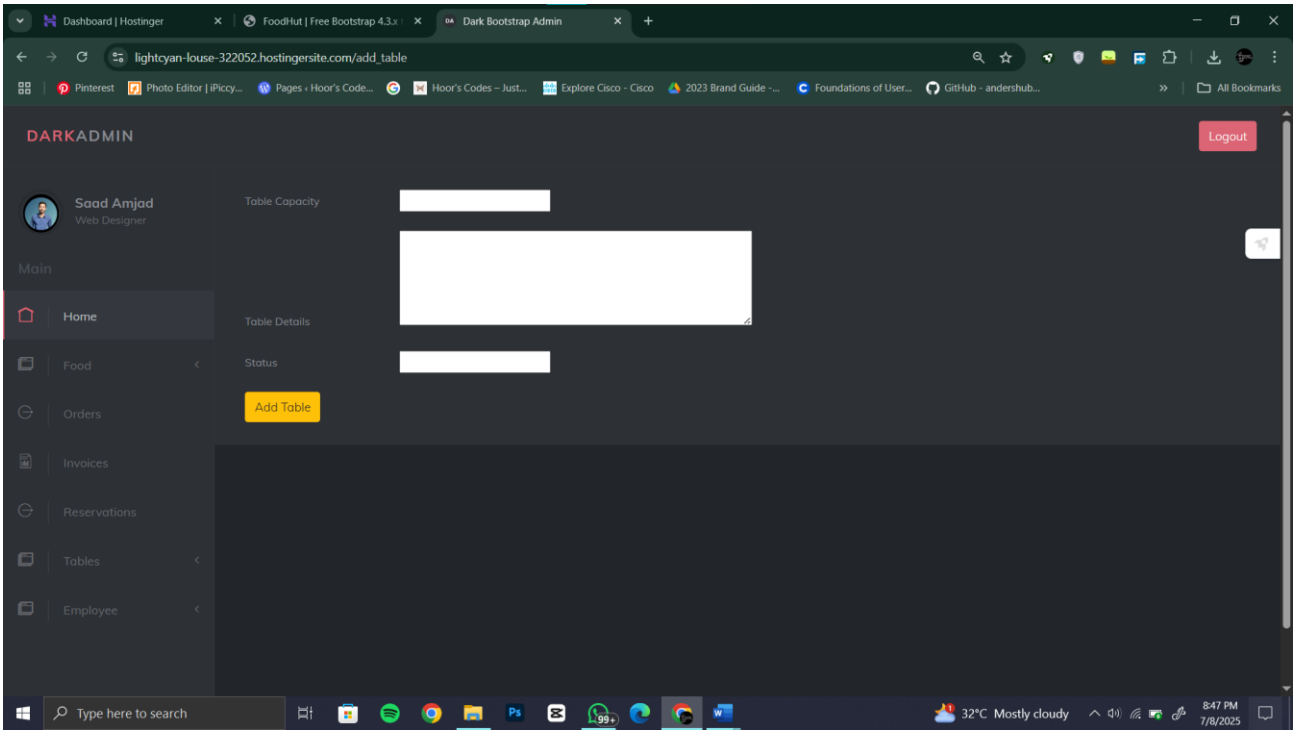
All invoices from admin panel:



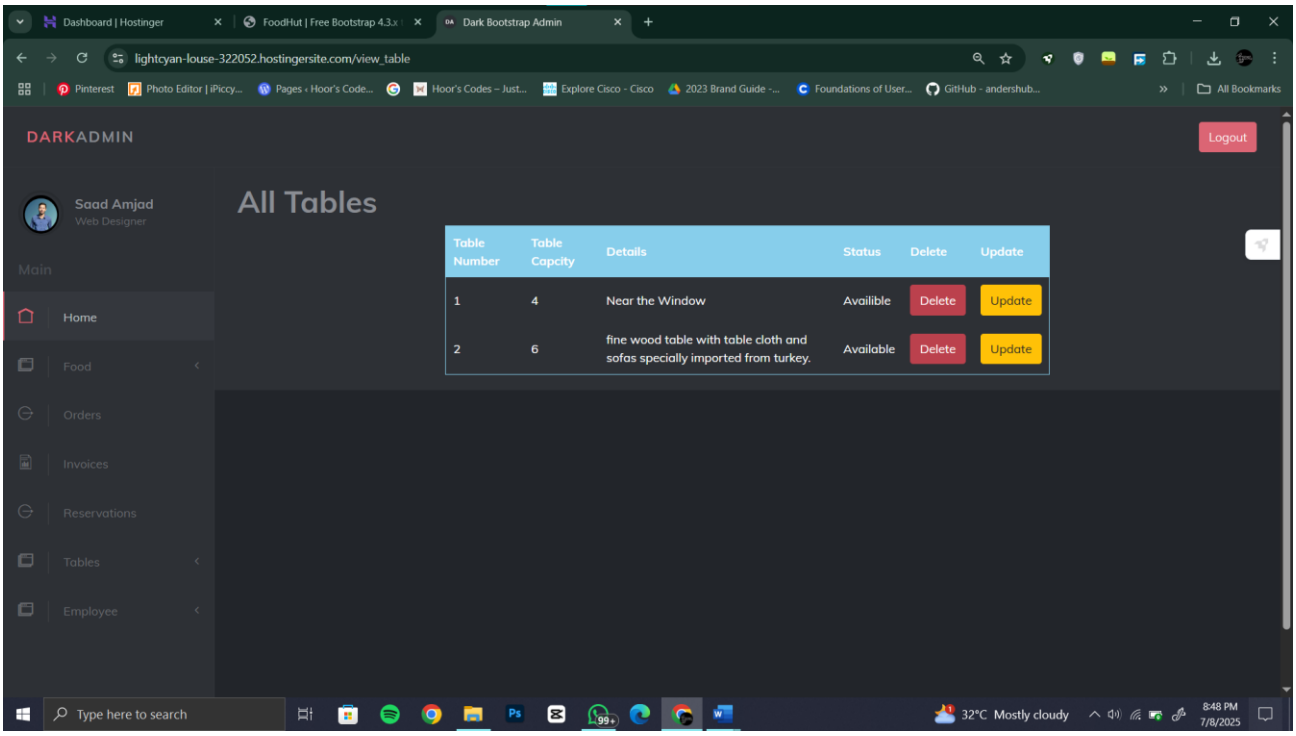
See all reservations from admin panel:



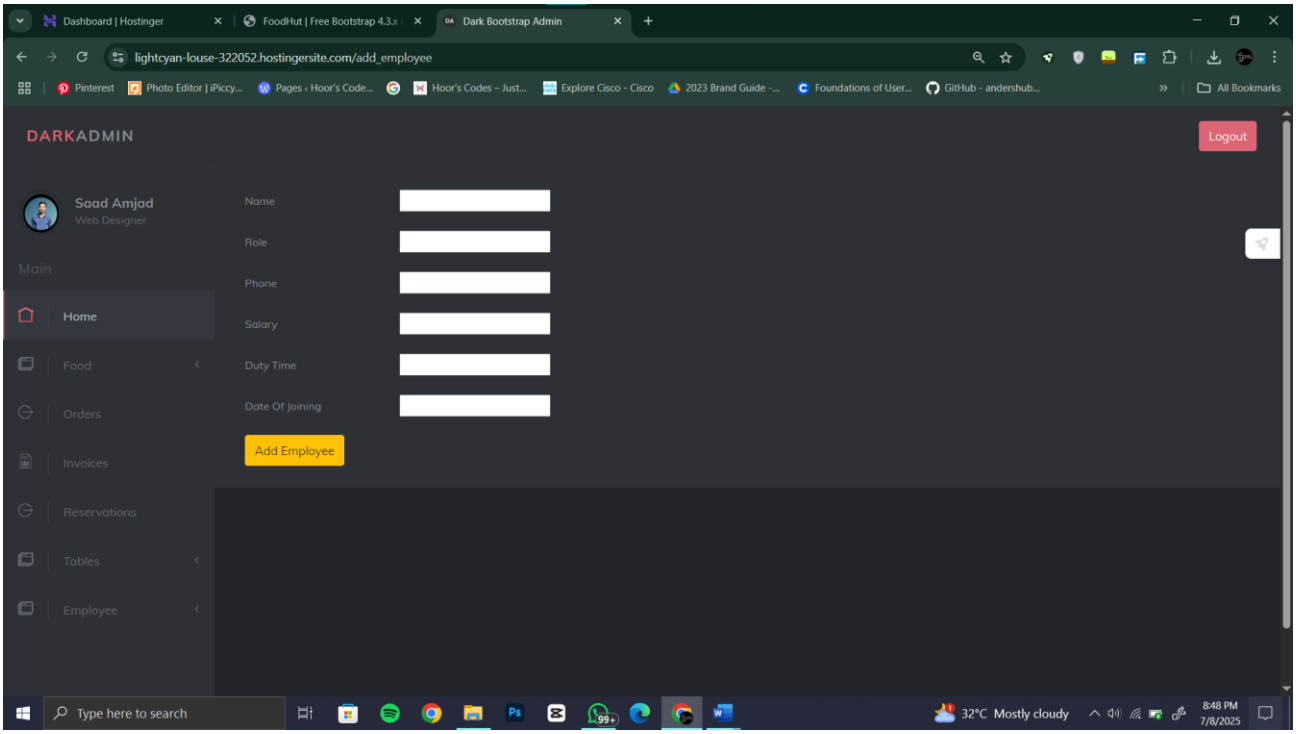
Add table at restaurant from admin panel:



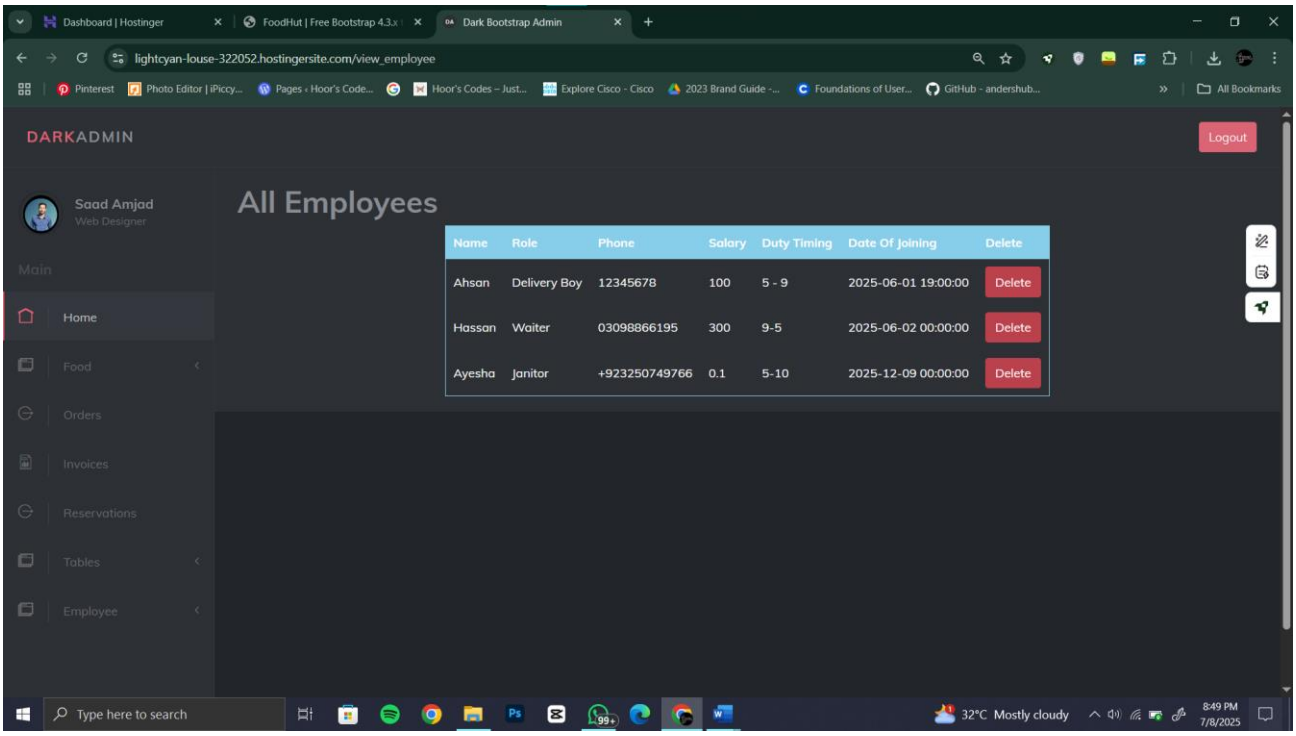
View all available tables from admin panel:



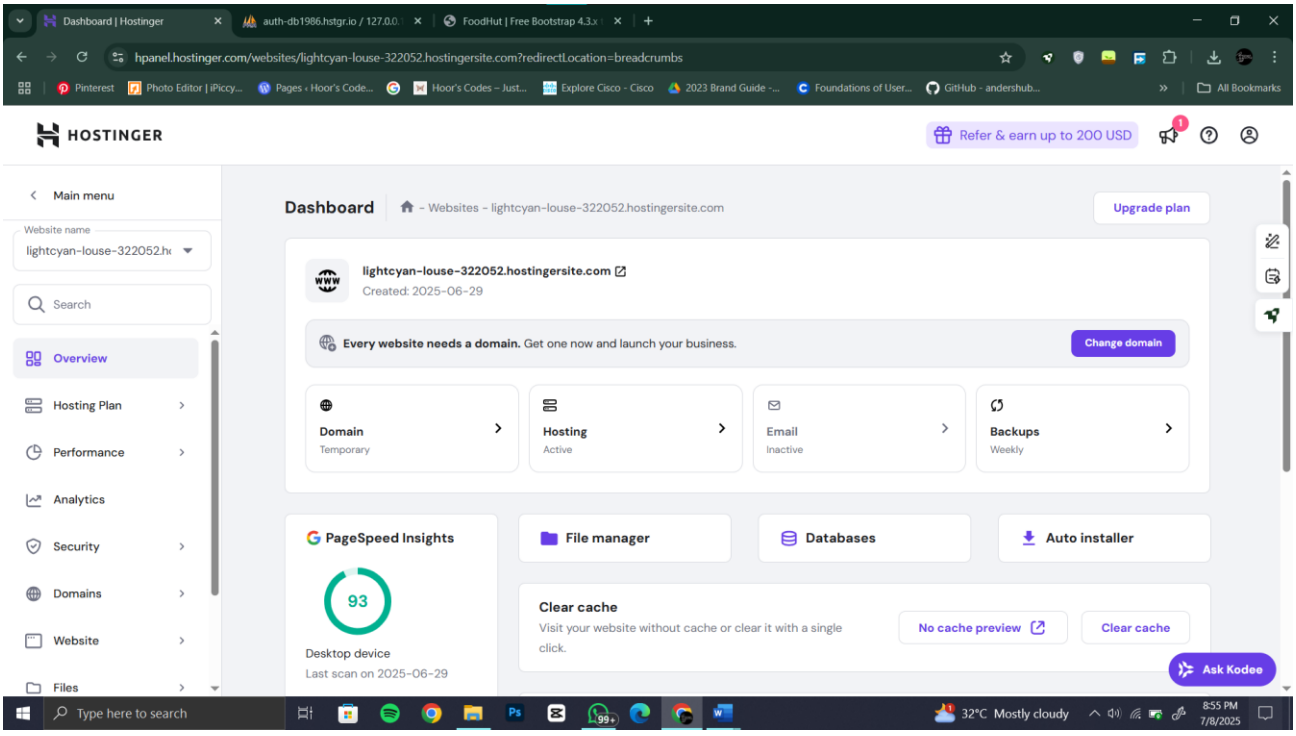
Add employee in our restaurant from admin panel:

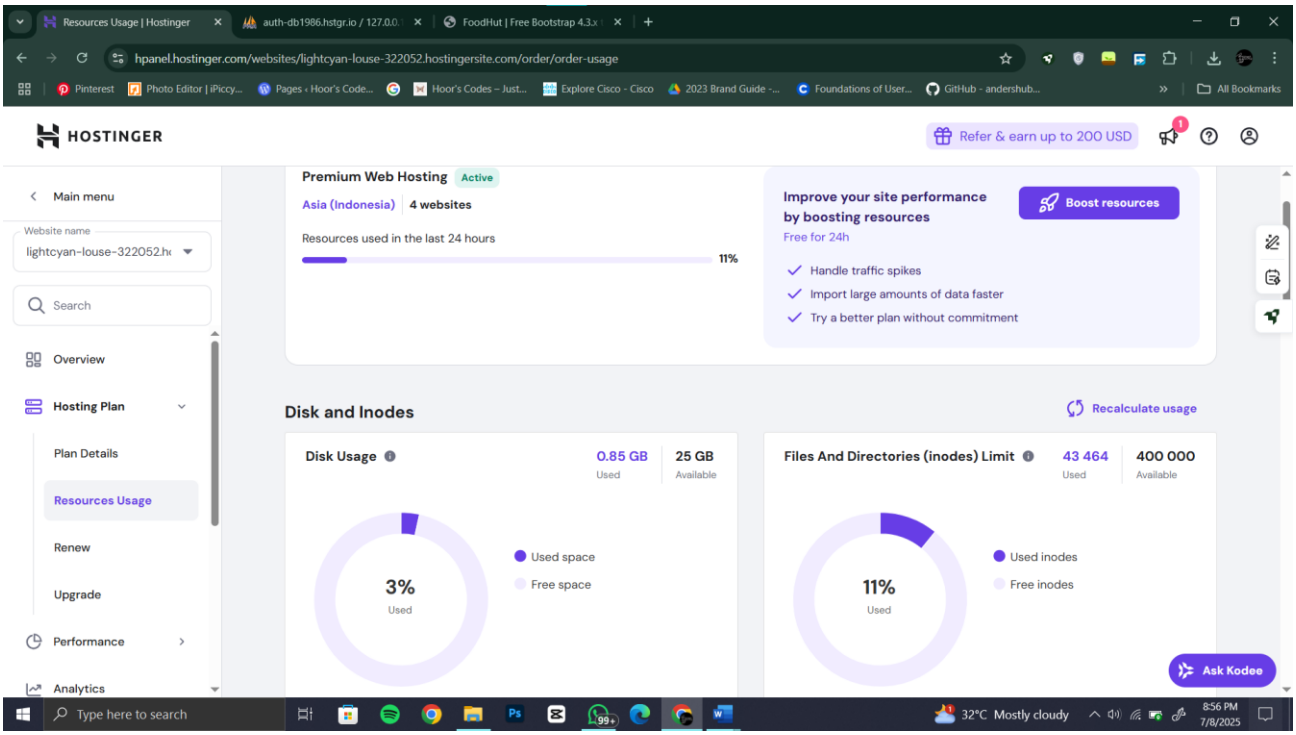


View all employees from admin panel:



Website deployment dashboard :





Website URL:
https://lightcyan-louse-322052.hostingersite.com/

Database credentials:

DB_DATABASE= u773711032_restaurant

DB_USERNAME= u773711032_kamiladmin

DB_PASSWORD=" Pesho@499"

Admin login credentials:

Email : admin@gmail.com

Password : 12345678

Conclusion

The Restaurant Management System simplifies restaurant operations and enhances the customer experience. With a modular architecture and modern technologies, it can be extended further by integrating features like online payments, notification emails, or multi-branch support.

References:

- ChatGPT
- Youtube
- Draw.io
- Cursor AI
- Claude