



Università degli studi di Messina
Dipartimento di Scienze Matematiche, Informatiche,
Scienze Fisiche e Scienze della Terra

Course: Data Analysis
Project on “WEB PROGRAMMING”

Project name: “Restaurant website”

A.A 2024/2025

Professor:
Armando Ruggeri

Student:
Baimurtov Bereke (551471)

Content:

- 1. Introduction**
- 2. Project Objectives**
- 3. Technologies Used**
- 4. Features Implemented**
- 5. Database**
- 6. Conclusion**
- 7. Appendices**

Introduction

The Restaurant Website Project, titled **Golden Spoon**, is a dynamic and interactive web application designed to enhance the customer and administrative experience for a modern restaurant. This project aims to bridge the gap between the restaurant and its patrons by offering a seamless platform for menu browsing, table reservations, and efficient administrative management.

The website was developed using a combination of frontend and backend technologies, including Flask, HTML5, CSS3, and MySQL, to deliver a responsive, visually appealing, and functional interface. With features tailored to meet both customer needs and administrative efficiency, the site embodies a user-friendly design paired with robust functionality.

Key functionalities include:

- **Customer Experience:** Guests can explore a categorized menu, reserve tables with ease, and learn about the restaurant's offerings and location.
- **Administrative Tools:** A secure admin panel allows restaurant staff to manage the menu, handle reservations, and ensure smooth operations.
- **Dynamic Design:** Integrated animations (via AOS library) and responsive layouts provide a polished, professional, and engaging user experience.

Project Objectives

The primary objectives of the **Golden Spoon** restaurant website project are as follows:

1. Enhance Customer Experience:

- Provide customers with a visually appealing and user-friendly platform to explore the restaurant's menu.
- Facilitate seamless table reservations through an intuitive booking system.
- Deliver essential restaurant information, including contact details, location, and operating hours.

2. Streamline Administrative Processes:

- Develop a secure admin panel for managing menu items, including the ability to add, edit, and delete dishes.
- Enable the management of customer reservations with ease and efficiency.
- Improve restaurant workflow by integrating technology into daily operations.

3. Create a Responsive and Dynamic Design:

- Build a fully responsive website that functions seamlessly across all devices, including desktops, tablets, and smartphones.
- Incorporate smooth animations and visual effects (using the AOS library) to enhance the aesthetic appeal of the site.

4. Facilitate Scalability and Customization:

- Use modular design principles to ensure that the website can easily accommodate additional features and functionality in the future.
- Allow for easy customization of branding elements, such as logos, colors, and text content.

5. Improve User Engagement:

- Encourage user interaction through dynamic navigation, engaging content, and visually appealing designs.
- Incorporate feedback mechanisms, such as a contact form, to foster customer communication.

6. Implement Secure Functionality:

- Ensure customer and administrative data is handled securely through best practices in coding and database management.
- Protect sensitive operations, such as admin panel access, with secure login features.

This project not only seeks to create a website for the restaurant but also aims to establish a modern digital presence that aligns with the restaurant's branding, enhances customer satisfaction, and boosts operational efficiency.

Technologies Used

The development of the **Golden Spoon** restaurant website project utilized a range of technologies to create a dynamic, user-friendly, and responsive platform. Below is an overview of the key technologies:

Frontend Technologies

1. HTML5:

- Used to structure the content of the website.
- Includes elements for headings, paragraphs, forms, images, and links.
- Provides semantic tags for better accessibility and SEO.

2. CSS3:

- Used for styling and layout.
- Key features:
- Custom styles for branding (colors, fonts, and layouts).
- Responsive design using media queries.
- Advanced effects like hover states, shadows, and animations.

3. JavaScript:

- Adds interactivity to the site.
- Examples:
- Smooth scrolling to categories.
- Show/hide "Back to Top" button.

4.AOS Library (Animate on Scroll):

- Used for smooth, modern animations as elements appear on the screen.
- Enhances user engagement and adds a polished feel to the site.

Backend Technologies

1.Python:

- Primary programming language for server-side logic.
- Framework: Flask:
- Lightweight framework for building web applications.
- Handles routing, form submission, and interaction with the database.
- Dynamic generation of content using Jinja2 templates.

2.Jinja2:

- Template engine used with Flask.

- Allows dynamic insertion of content into HTML, such as menu items and reservation data.

Database

1.MySQL:

Used to store and manage:

- Menu items (name, description, price, image, category).
- Customer reservations (name, phone number, date, and table number).
- Admin user data for secure login.

Managed via phpMyAdmin for easy database administration.

Development Tools

1.XAMPP:

- Provides a local development environment for running MySQL and testing the project locally.

2.VS Code:

- Primary code editor for writing HTML, CSS, Python, and JavaScript.
- Extensions for Flask development, debugging, and formatting.

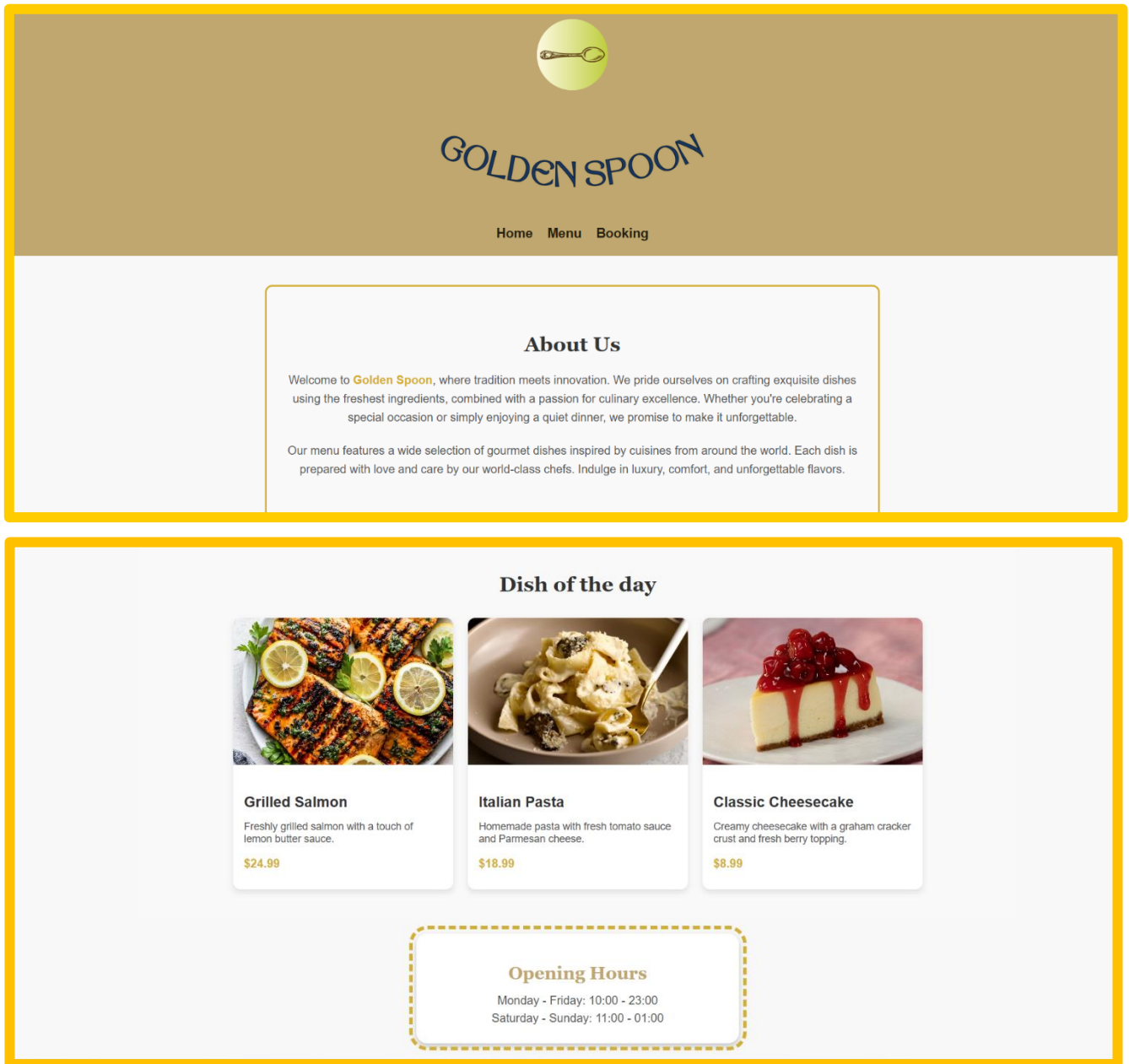
Features Implemented

1.Client-Side Features

These features are designed for restaurant customers, focusing on user experience and engagement.

1.1 Home Page:

Description: Displays a welcoming introduction to the restaurant.



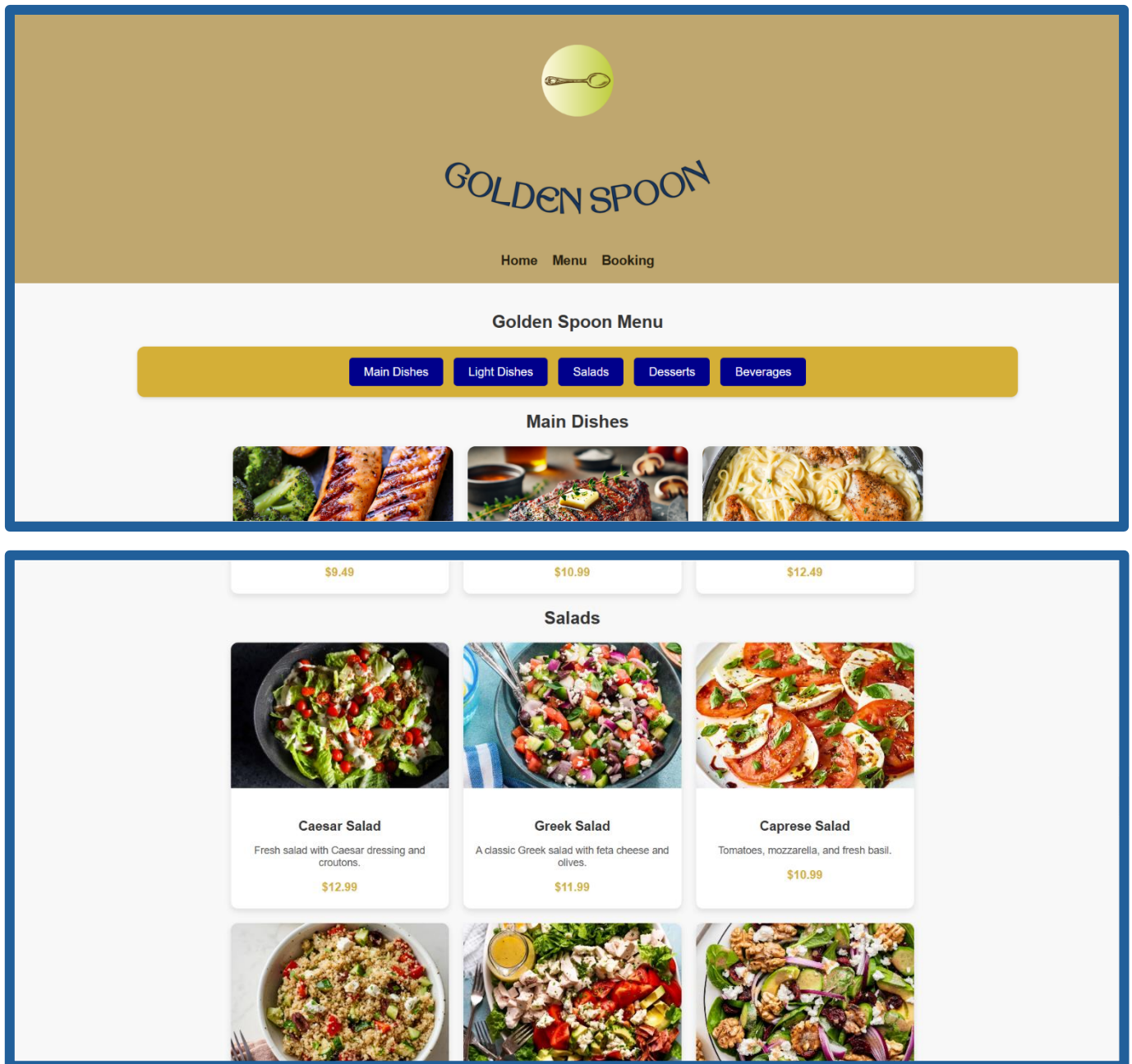
Key Elements:

- A brief "About Us" section with an animated title and background visuals.
- Featured dishes section with images and descriptions.
- Restaurant location with a map or image of the address.

- Social media links for Instagram, Facebook, and other platforms.

1.2 Menu Page:

Description: Allows users to browse the menu, categorized by dish types.



Key Elements:

- Categories like Main Dishes, Light Dishes, Salads, Desserts, and Beverages.
- Navigation buttons to quickly jump between categories.
- Dynamic data rendering for menu items, including name, price, description, and image.
- Hover animations for enhanced interactivity.

1.3 Table Booking Page:

Description: Provides a form for customers to book tables at the restaurant.

Home Menu Booking

Make a Reservation

Your Name:

Phone Number:

Date and Time:

Number of Guests:

Book Now

Key Elements:

- Input fields for name, phone number, date, time, and number of guests.
- Validation to ensure all fields are correctly filled out.
- Success or error messages after form submission.

1.4 Responsive Design:

Description: The site adapts to various devices, including desktops, tablets, and smartphones.

Key Elements:

- Media queries for different screen sizes.
- Responsive navigation bar and menu layout.

1.5 Animations (AOS Library):

Description: Adds smooth animations to various elements as they appear on the screen.

Key Elements:

- Effects like fade-up, zoom-in, and slide-right on menu items, images, and section headings.

2. Admin-Side Features

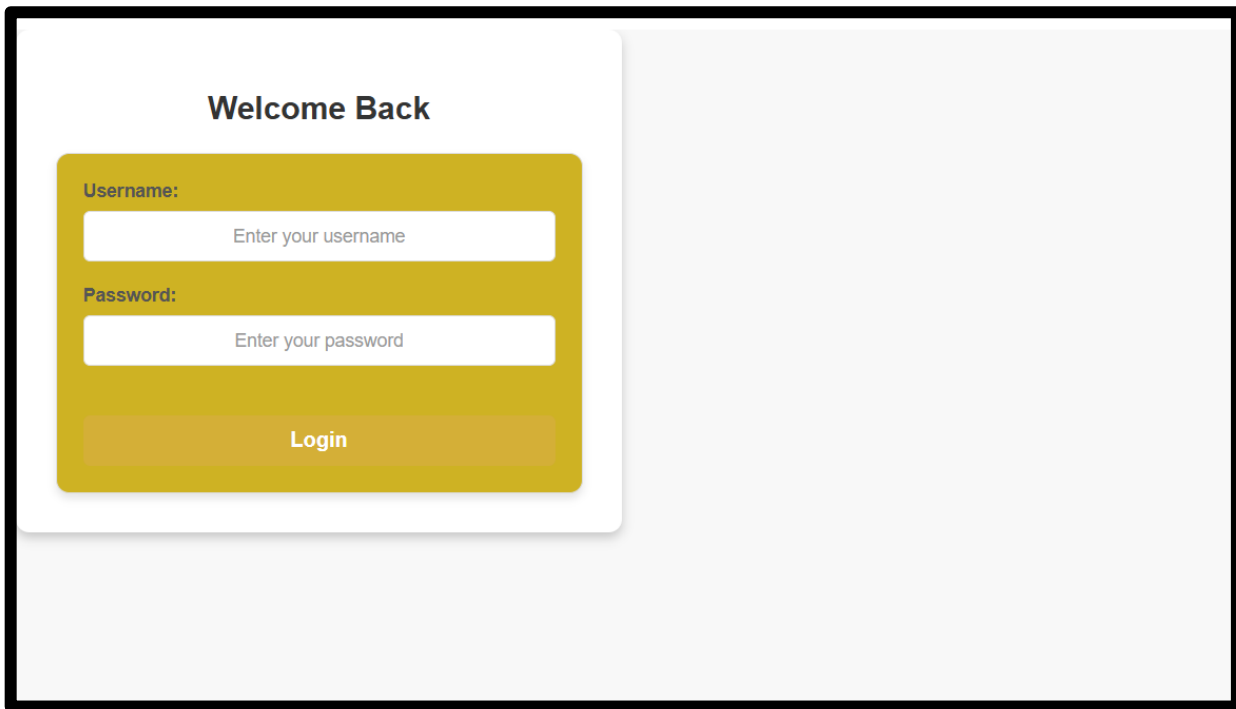
These features are tailored for restaurant administrators to manage the site efficiently.

2.1 Admin Login:

Description: Provides a secure login page for administrators.

Key Elements:

- Username and password fields with validation.

A login form titled "Welcome Back" with a yellow header. It contains two input fields: "Username:" with placeholder text "Enter your username" and "Password:" with placeholder text "Enter your password". Below the fields is a yellow "Login" button.

Welcome Back

Username:

Enter your username

Password:

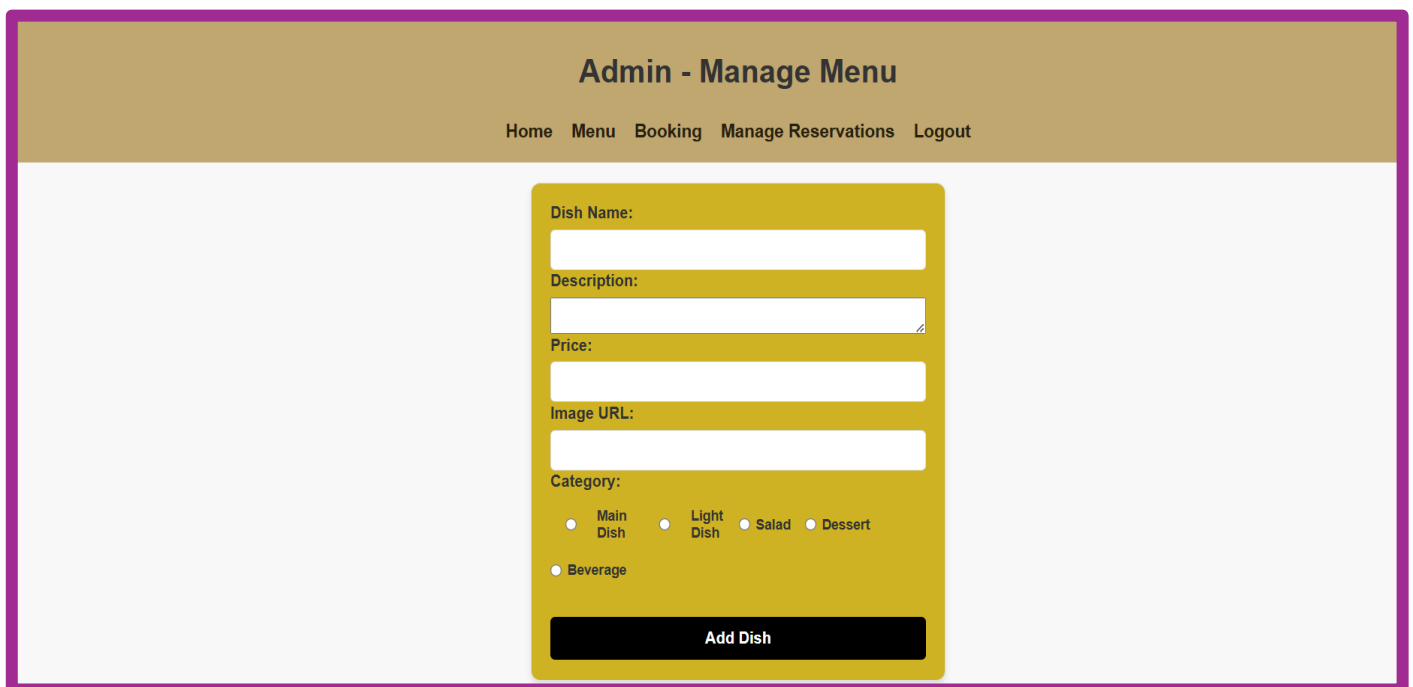
Enter your password

Login

- Backend logic to verify credentials.
- Sessions for secure access to the admin panel.

2.2 Menu Management:

Description: Enables administrators to manage the restaurant's menu dynamically.

An admin panel titled "Admin - Manage Menu" with a brown header. It has a navigation bar with links: Home, Menu, Booking, Manage Reservations, and Logout. The main content area contains a yellow form for adding a dish. The form has fields for "Dish Name:", "Description:", "Price:", and "Image URL:". Below these is a "Category:" section with radio buttons for "Main Dish", "Light Dish", "Salad", "Dessert", and "Beverage". At the bottom is a black "Add Dish" button.

Admin - Manage Menu

Home Menu Booking Manage Reservations Logout

Dish Name:

Description:

Price:

Image URL:

Category:

☐ Main Dish ☐ Light Dish ☐ Salad ☐ Dessert

☐ Beverage

Add Dish

Key Elements:

- Add new dishes with name, description, price, category, and image URL.
- Edit existing menu items.
- Delete dishes no longer offered.
- Dynamic table displaying the current menu.

Current Menu					
ID	Name	Description	Price	Category	Actions
16	Grilled Salmon	Freshly grilled salmon with lemon butter sauce.	\$24.99	Main Dish	<button>Delete</button>
17	Beef Steak	Juicy beef steak grilled to perfection.	\$29.99	Main Dish	<button>Delete</button>
18	Chicken Alfredo	Creamy pasta with chicken and Alfredo sauce.	\$19.99	Main Dish	<button>Delete</button>
19	Lamb Chops	Tender lamb chops with rosemary seasoning.	\$34.99	Main Dish	<button>Delete</button>
20	Bruschetta	Toasted bread with fresh tomatoes, basil, and olive oil.	\$8.99	Light Dish	<button>Delete</button>
21	Spring Rolls	Crispy spring rolls filled with vegetables.	\$6.99	Light Dish	<button>Delete</button>
22	Garlic Bread	Warm garlic bread with herbs.	\$5.49	Light Dish	<button>Delete</button>
23	Stuffed Mushrooms	Mushrooms stuffed with cheese and breadcrumbs.	\$7.99	Light Dish	<button>Delete</button>
24	Caesar Salad	Fresh salad with Caesar dressing and croutons.	\$12.99	Salad	<button>Delete</button>

2.3 Reservation Management:

Description: Provides a list of all customer reservations for review.

Manage Reservations					
Home Menu Booking Manage Menu Logout					
Current Reservations					
ID	Customer Name	Phone	Reservation Date	Table Number	Actions
6	Bereke Baimuratov	+393283472416	2025-01-21 13:30:00	6	<button>Edit</button> <button>Delete</button>
10	Dair	+393283472416	2025-01-15 20:53:00	3	<button>Edit</button> <button>Delete</button>
12	Akmaral	+393283472416	2025-01-16 15:50:00	5	<button>Edit</button> <button>Delete</button>
13	Anelya	+393283472416	2025-01-16 15:41:00	4	<button>Edit</button> <button>Delete</button>
15	Daiana	+393283472416	2025-01-16 20:41:00	1	<button>Edit</button> <button>Delete</button>
16	Rustem	+393283472416	2025-01-16 19:42:00	9	<button>Edit</button> <button>Delete</button>
17	Atai	+393283472416	2025-01-16 21:42:00	5	<button>Edit</button> <button>Delete</button>

© 2025 Golden Spoon Admin Panel

Key Elements:

- View reservations with details like name, phone, date, time, and guests.
- Delete past or invalid reservations.

2.4 Navigation and User Interface:

Description: Admin panel navigation is intuitive and consistent.

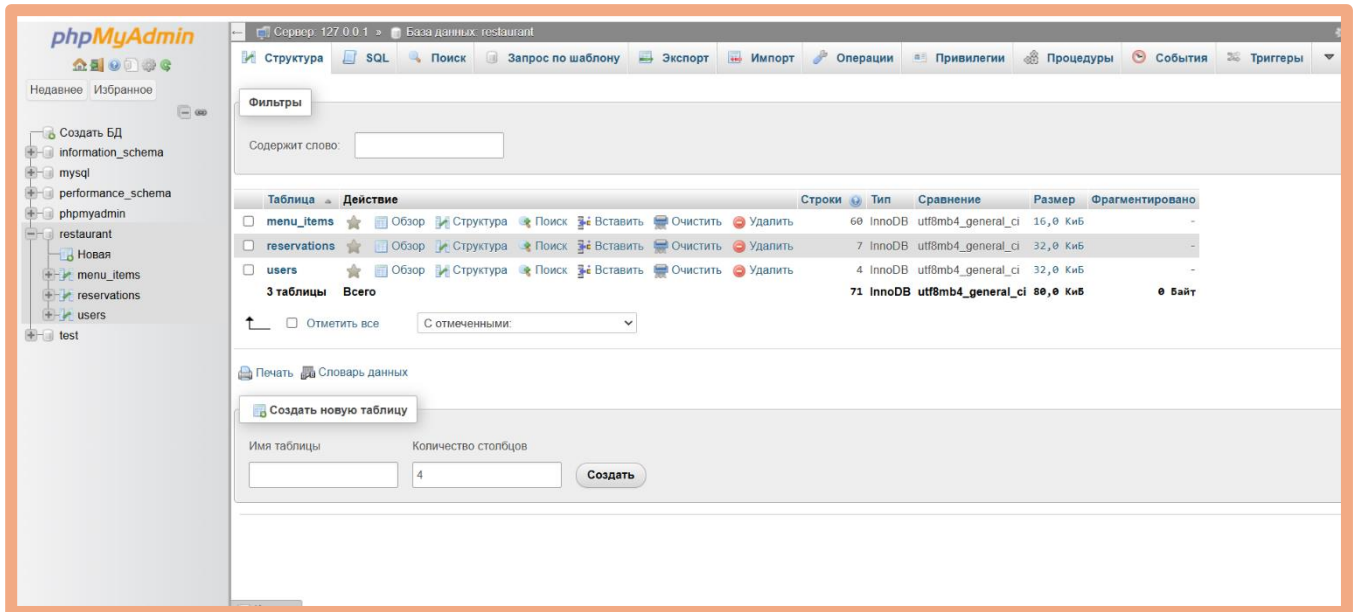
Key Elements:

- Separate sections for menu and reservation management.
- Button-based navigation for key admin tasks.

3. Database Integration

3.1 Dynamic Content Loading:

- Menu items and reservations are fetched from a MySQL database.
- Data is updated in real-time based on admin actions (add/edit/delete).



3.2 Form Submission:

- Booking forms send data directly to the database.
- Error handling for invalid or incomplete submissions.

4. Aesthetic Features

4.1 Custom Styling:

- Golden and beige tones for a luxurious restaurant feel.
- Smooth hover effects and shadows for buttons and images.
- Stylish footer with contact information and social media icons.

4.2 "Back to Top" Button:

- Smooth scrolling to the top of the page when clicked.

This combination of features ensures both customers and administrators have a seamless and enjoyable experience while using the site. Let me know if you need more details on any specific feature.

Database

The **Golden Spoon** project uses a MySQL database to store and manage all the essential data required for the website's functionality. Below is a detailed breakdown of the database design:

1. Database Overview

The database contains the following tables:

1. menu_items: Stores details about the restaurant's menu items.
2. reservations: Manages customer table reservations.
3. users: Handles administrator login credentials.

2. Table Structures

1. menu_items

- Purpose: Stores information about each dish in the menu.
- Columns:

Column Name	Data Type	Description
id	INT (Primary Key)	Unique ID for each menu item.
name	VARCHAR(100)	Name of the dish.
description	TEXT	Detailed description of the dish.
price	DECIMAL(10,2)	Price of the dish.
image_url	VARCHAR(255)	Path to the dish's image.
category	VARCHAR(50)	Category (e.g., Main Dish, Dessert).

2. reservations

- Purpose: Stores information about table reservations made by customers.
- Columns:

Column Name	Data Type	Description
id	INT (Primary Key)	Unique ID for each reservation.
customer_name	VARCHAR(100)	Name of the customer.
phone	VARCHAR(15)	Customer's phone number.
reservation_date	DATETIME	Date and time of the reservation.
table_number	INT	Number of guests.

3. users

- Purpose: Stores administrator login information.
- Columns:

Column Name	Data Type	Description
id	INT (Primary Key)	Unique ID for each admin user.
Username	VARCHAR(50)	Administrator's username.
Password	TEXT	Hashed password for security.
is_admin	BOOLEAN	Indicates if the user is an admin.

3. Relationships

The tables are designed to work independently for simplicity, without complex foreign key relationships.

Example:

- The menu_items table handles menu-specific data.
- The reservations table operates independently, storing customer bookings.

4. SQL Queries

Here are some examples of queries used in the project:

1. Insert Data:

- Add a new dish to the menu:

```
INSERT INTO menu_items (name, description, price, image_url, category)
VALUES ('Grilled Salmon', 'Freshly grilled salmon.', 24.99,
'images/salmon.jpg', 'Main Dish');
```

- Add a new reservation:

```
INSERT INTO reservations (customer_name, phone, reservation_date,
table_number)
VALUES ('John Doe', '555-1234567', '2025-12-20 18:30:00', 4);
```

2. Retrieve Data:

- Get all menu items:

```
SELECT * FROM menu_items;
```

- Get reservations for a specific date:

```
SELECT * FROM reservations WHERE reservation_date LIKE '2025-12-20%';
```

3. Update Data:

- Update a dish's price:

```
UPDATE menu_items SET price = 26.99 WHERE id = 1;
```

4. Delete Data:

- Remove a reservation:

```
DELETE FROM reservations WHERE id = 2;
```

5. Database Management

- phpMyAdmin: Used for manual management of the database (e.g., creating tables, running queries).
- Flask MySQL Integration: Database queries and updates are handled programmatically via Flask, ensuring dynamic interaction.

Conclusion

The **Golden Spoon** Restaurant Website Project successfully achieved its primary objectives, providing a functional and aesthetically pleasing platform for both customers and administrators. This project highlights the seamless integration of frontend, backend, and database technologies to deliver a responsive, dynamic, and user-friendly experience.

Key Achievements:

1.Customer-Focused Features:

- The website offers an engaging and easy-to-navigate interface for browsing the restaurant menu and making table reservations.
- A responsive design ensures that the site works seamlessly across desktops, tablets, and mobile devices.
- Smooth animations and transitions enhance the visual appeal and overall user experience.

2.Administrative Efficiency:

- A secure admin panel allows restaurant staff to manage menu items and reservations with ease.
- The use of Flask and MySQL ensures real-time updates and efficient data management.

3.Technical Excellence:

- Full integration of Flask for routing, dynamic content rendering (Jinja2), and database connectivity.
- Implementation of MySQL to store and organize critical restaurant data, such as menu items, reservations, and user credentials.
- Deployment of animations through the AOS library to enhance the professional appearance of the website.

4.Scalability and Customization:

- The modular design of the website makes it easy to extend and customize, allowing for future features like user reviews, loyalty programs, or multi-language support.

Lessons Learned:

This project provided valuable insights into:

- Building and managing a full-stack web application using Flask.
- Working with databases to handle dynamic content.
- Designing responsive and user-friendly interfaces.

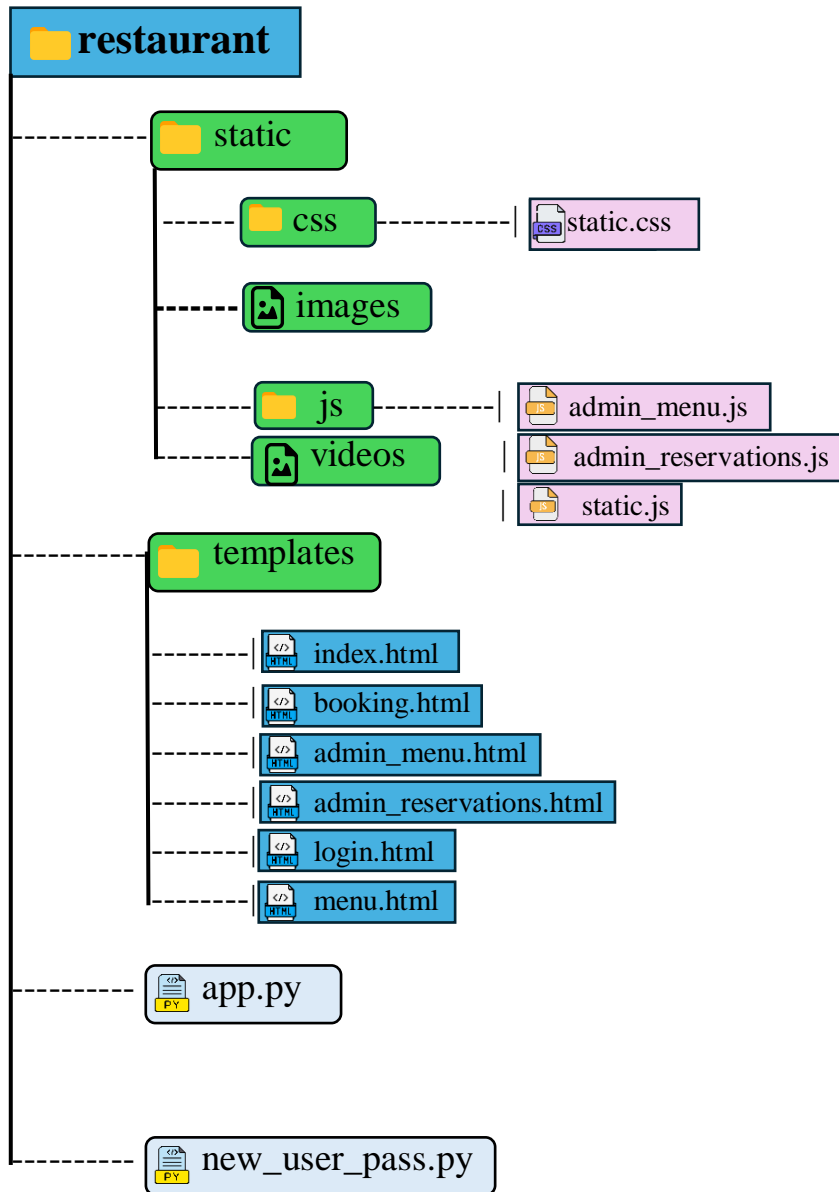
Future Enhancements:

- Adding a review system where customers can leave feedback on their dining experience.
- Implementing a loyalty program for returning customers.
- Expanding functionality for administrators, such as generating sales reports or advanced reservation management.
- Integrating payment options for reservations or online orders.

In conclusion, the **Golden Spoon** Restaurant Website Project serves as a practical demonstration of modern web development techniques. It combines functionality and aesthetics to create a platform that meets the needs of both customers and restaurant staff, establishing a strong foundation for further enhancements.

Appendices

Structure of files:





app.py:

```
from flask import Flask, render_template, request, jsonify, session, redirect, url_for
from flask_mysql import MySQL
from werkzeug.security import generate_password_hash, check_password_hash
from flask_session import Session
import datetime

app = Flask(__name__)

# Конфигурация базы данных
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '' # Укажите ваш пароль, если установлен
app.config['MYSQL_DB'] = 'restaurant'
app.config['SECRET_KEY'] = 'your_secret_key_here'

# Конфигурация сессий
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_PERMANENT'] = False
Session(app)

mysql = MySQL(app)

# Главная страница
@app.route('/')
def home():
    return render_template('index.html')

# Страница меню
@app.route('/menu')
def menu():
    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM menu_items")
    menu_items = cursor.fetchall()
    return render_template('menu.html', menu=menu_items)

@app.route('/booking', methods=['GET', 'POST'])
def booking():
    if request.method == 'POST':
        customer_name = request.form.get('customer_name')
        phone = request.form.get('phone')
        reservation_date = request.form.get('reservation_date')
        table_number = request.form.get('table_number')

        if not all([customer_name, phone, reservation_date, table_number]):
            return render_template('booking.html', error="All fields are required!")

    try:
        cursor = mysql.connection.cursor()
        cursor.execute("""
            INSERT INTO reservations (customer_name, phone, reservation_date, table_number)
            VALUES (%s, %s, %s, %s)
            """, (customer_name, phone, reservation_date, table_number))
        mysql.connection.commit()
```

```

        return render_template('booking.html', success="Your table has been successfully booked!")
    except Exception as e:
        return render_template('booking.html', error="An error occurred while booking. Please try again.")

    return render_template('booking.html')

@app.route('/admin/menu', methods=['GET', 'POST'])
def admin_menu():
    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM menu_items")
    menu_items = cursor.fetchall()
    return render_template('admin_menu.html', menu=menu_items)

# Админская панель: управление бронированиями
@app.route('/admin/reservations')
def admin_reservations():
    if not session.get('is_admin'):
        return redirect(url_for('login_page'))

    cursor = mysql.connection.cursor()
    cursor.execute("SELECT * FROM reservations")
    reservations = cursor.fetchall()
    return render_template('admin_reservations.html', reservations=reservations)

# Страница логина
@app.route('/login', methods=['GET', 'POST'])
def login_page():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

        cursor = mysql.connection.cursor()
        cursor.execute("SELECT id, password, is_admin FROM users WHERE username = %s", (username,))
        user = cursor.fetchone()

        if user and check_password_hash(user[1], password):
            # Устанавливаем данные пользователя в сессию
            session['user_id'] = user[0]
            session['is_admin'] = user[2]
            return redirect(url_for('admin_menu') if user[2] else url_for('home'))
        else:
            return render_template('login.html', error="Invalid credentials")

    return render_template('login.html')

# Выход из системы
@app.route('/logout', methods=['GET'])
def logout():
    session.clear() # Очищаем сессию
    return redirect(url_for('home'))

@app.route('/admin/menu/add', methods=['POST'])
def add_dish():
    name = request.form.get('name')
    description = request.form.get('description')

```

```

price = request.form.get('price')
image_url = request.form.get('image_url')
category = request.form.get('category') # Получаем категорию

# Проверяем, выбрана ли категория
if not category:
    return jsonify({"error": "Category is required!"}), 400

# Добавляем новое блюдо в базу данных
cursor = mysql.connection.cursor()
cursor.execute(
    "INSERT INTO menu_items (name, description, price, image_url, category) VALUES (%s, %s, %s, %s, %s)",
    (name, description, price, image_url, category)
)
mysql.connection.commit()
return jsonify({"message": "Dish added successfully!"})

@app.route('/admin/reservations/delete', methods=['POST'])
def delete_reservation():
    if not session.get('is_admin'):
        return jsonify({"error": "Access forbidden: Admins only"}), 403

    data = request.json
    reservation_id = data.get('id')

    if not reservation_id:
        return jsonify({"error": "Reservation ID is required"}), 400

    cursor = mysql.connection.cursor()
    cursor.execute("DELETE FROM reservations WHERE id = %s", (reservation_id,))
    mysql.connection.commit()
    return jsonify({"message": "Reservation deleted successfully!"})

@app.route('/admin/reservations/edit', methods=['POST'])
def edit_reservation():
    if not session.get('is_admin'):
        return jsonify({"error": "Access forbidden: Admins only"}), 403

    data = request.json
    reservation_id = data.get('id')
    customer_name = data.get('customer_name')
    phone = data.get('phone')
    reservation_date = data.get('reservation_date')
    table_number = data.get('table_number')

    if not all([reservation_id, customer_name, phone, reservation_date, table_number]):
        return jsonify({"error": "All fields are required"}), 400

    cursor = mysql.connection.cursor()
    cursor.execute("""
        UPDATE reservations
        SET customer_name = %s, phone = %s, reservation_date = %s, table_number = %s
        WHERE id = %s
        """, (customer_name, phone, reservation_date, table_number, reservation_id))
    mysql.connection.commit()

```

```

return jsonify({"message": "Reservation updated successfully!"})

@app.route('/admin/menu/delete/<int:item_id>', methods=['POST'])
def delete_menu_item(item_id):
    cursor = mysql.connection.cursor()
    cursor.execute("DELETE FROM menu_items WHERE id = %s", (item_id,))
    mysql.connection.commit()
    return redirect(url_for('admin_menu'))

# Главный блок для запуска сервера
if __name__ == '__main__':
    app.run(debug=True)

```



index.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Golden Spoon</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/aos/2.3.4/aos.css"> <!-- Animate on Scroll -->
</head>
<body>
    <header>
        <div class="logo" data-aos="zoom-out">
            
        </div>
        <div class="name_res" data-aos="zoom-out">
            
        </div>
        <nav>
            <a href="{{ url_for('home') }}">Home</a>
            <a href="{{ url_for('menu') }}">Menu</a>
            <a href="{{ url_for('booking') }}">Booking</a>
        </nav>
    </header>
    <main>
        <section id="about" data-aos="fade-up">
            <h2>About Us</h2>
            <div class="about-container">
                <p>
                    Welcome to <strong>Golden Spoon</strong>, where tradition meets innovation.
                    We pride ourselves on crafting exquisite dishes using the freshest ingredients,
                    combined with a passion for culinary excellence. Whether you're celebrating a special
                    occasion or simply enjoying a quiet dinner, we promise to make it unforgettable.
                </p>
                <p>
                    Our menu features a wide selection of gourmet dishes inspired by cuisines from around the world.
                    Each dish is prepared with love and care by our world-class chefs. Indulge in luxury, comfort, and
                    unforgettable flavors.
                </p>
            </div>
        </section>
        <section id="kitchen" data-aos="zoom-out">

```

```
<h2>Our Kitchen in Action</h2>
<div class="kitchen-container">
  <video autoplay muted loop playsinline class="kitchen-video">
    <source src="{ { url_for('static', filename='videos/kitchen.mp4') } }" type="video/mp4">
    Your browser does not support the video tag.
  </video>
</div>
</section>
<section id="celebrity-reviews">
  <h2>Stars Who Chose Us</h2>
  <div class="reviews-container">
    <!-- Звезда 1 -->
    <div class="review-card">
      
      <h3>Elizabeth Olsen</h3>
      <p class="rating">★★★★★</p>
      <p>"The food and ambiance were absolutely delightful. Highly recommend!"</p>
    </div>
    <!-- Звезда 2 -->
    <div class="review-card">
      
      <h3>Armando Ruggeri</h3>
      <p class="rating">★★★★★</p>
      <p>"This restaurant was opened by my best student. The dishes are the most delicious in the world!"</p>
    </div>
    <!-- Звезда 3 -->
    <div class="review-card">
      
      <h3>Taylor Swift</h3>
      <p class="rating">★★★★★</p>
      <p>"This restaurant exceeded all my expectations. Loved it!"</p>
    </div>
    <!-- Звезда 4 -->
    <div class="review-card">
      
      <h3>Robert Downey Jr.</h3>
      <p class="rating">★★★★★</p>
      <p>"Unforgettable dining experience. Hats off to the chefs!"</p>
    </div>
  </div>
</section>
<section id="menu-highlight" data-aos="zoom-out">
  <h2>Dish of the day</h2>
  <div class="menu-highlight-container">
    <div class="menu-card">
      
      <div class="menu-info">
        <h3>Grilled Salmon</h3>
        <p>Freshly grilled salmon with a touch of lemon butter sauce.</p>
        <p class="price">$24.99</p>
      </div>
    </div>
    <div class="menu-card">
      
      <div class="menu-info">
        <h3>Italian Pasta</h3>
        <p>Homemade pasta with fresh tomato sauce and Parmesan cheese.</p>
        <p class="price">$18.99</p>
      </div>
    </div>
  </div>
</div>
```

```

<div class="menu-card">
  
  <div class="menu-info">
    <h3>Classic Cheesecake</h3>
    <p>Creamy cheesecake with a graham cracker crust and fresh berry topping.</p>
    <p class="price">$8.99</p>
  </div>
</div>
</div>
</div>
</section>
<section id="hours">
  <div class="hours-container">
    <h2>Opening Hours</h2>
    <p>Monday - Friday: 10:00 - 23:00 </p>
    <p>Saturday - Sunday: 11:00 - 01:00 </p>
  </div>
</section>
<section id="location" data-aos="zoom-out">
  <h2>Find Us</h2>
  <div class="location-container">
    
    <p>Salita Comunale Sperone, 31, 98100 Messina ME</p>
  </div>
</section>
</main>
<footer>
  <a href="{{ url_for('admin_menu') }}" class="admin-button">
    <a href="{{ url_for('login_page') }}" class="admin-button">Admin Panel</a>
  </a>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.3.4/aos.js"></script>
  <div class="footer-content">
    <div class="contact-info">
      <h4>Contact Us</h4>
      <p>Email: info@goldenspoon.com</p>
      <p>Phone: +39 (328) 3472416 </p>
      <p>Address: Salita Comunale Sperone, 31, 98100 Messina ME</p>
    </div>
    <div class="social-media">
      <h4>Follow Us</h4>
      <a href="https://instagram.com" target="_blank">
        
      </a>
      <a href="https://facebook.com" target="_blank">
        
      </a>
      <a href="https://x.com" target="_blank">
        
      </a>
    </div>
  </div>
  <p>&copy; 2025 Golden Spoon. All Rights Reserved.</p>
  <script>
    AOS.init({
      duration: 1200, // Длительность анимации (в миллисекундах)
      once: true // Анимация будет запускаться только один раз
    });
  </script>
</footer>
</body>
</html>

```




admin_reservation.js:

```
function deleteReservation(id) {
  if (confirm("Are you sure you want to delete this reservation?")) {
    fetch('/admin/reservations/delete', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ id: id })
    })
    .then(response => response.json())
    .then(data => {
      alert(data.message || data.error);
      location.reload(); // Перезагрузка страницы после удаления
    })
    .catch(error => console.error('Error:', error));
  }
}

function editReservation(id, customer_name, phone, reservation_date, table_number) {
  const newName = prompt("Enter new customer name:", customer_name);
  const newPhone = prompt("Enter new phone number:", phone);
  const newDate = prompt("Enter new reservation date (YYYY-MM-DD HH:MM:SS):", reservation_date);
  const newTable = prompt("Enter new table number:", table_number);

  if (newName && newPhone && newDate && newTable) {
    fetch('/admin/reservations/edit', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        id: id,
        customer_name: newName,
        phone: newPhone,
        reservation_date: newDate,
        table_number: newTable
      })
    })
    .then(response => response.json())
    .then(data => {
      alert(data.message || data.error);
      location.reload(); // Перезагрузка страницы после обновления
    })
    .catch(error => console.error('Error:', error));
  }
}
```