

Advanced Python Programming (APP)

Course Code (CC) – CSI – 3007

Laboratory (Lab) Class Live Day To Day Assessment Assignment

Web App Development - Flask

Name :- Dharshan Raj P A

Register Number :- 22MIC0073

Slot :- A2 + L7 + L8 + L27 + L28

Faculty:-Dr. Prof. Sharmila Banu K

Semester:- Fall Semester (2024-2025)

1)Food Delivery App:-

Projects-J_Component\Flask\static\css\style.css:-

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

:root {
    --primary-color: #ff6b35;
    --secondary-color: #f7931e;
    --dark-color: #2c3e50;
    --light-color: #ecf0f1;
    --success-color: #27ae60;
    --danger-color: #e74c3c;
    --warning-color: #f39c12;
    --info-color: #3498db;
    --white: #ffffff;
    --gray: #95a5a6;
    --dark-gray: #7f8c8d;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    line-height: 1.6;
    color: var(--dark-color);
    background-color: #f8f9fa;
    min-height: 100vh;
```

```
        display: flex;
        flex-direction: column;
    }

    .container {
        max-width: 1200px;
        margin: 0 auto;
        padding: 0 20px;
        width: 100%;
    }

/* Navbar */
.navbar {
    background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
    color: var(--white);
    padding: 1rem 0;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    position: sticky;
    top: 0;
    z-index: 1000;
}

.navbar .container {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.logo {
    font-size: 1.5rem;
    font-weight: bold;
    color: var(--white);
    text-decoration: none;
    display: flex;
    align-items: center;
    gap: 0.5rem;
}

.nav-menu {
    display: flex;
    list-style: none;
    gap: 2rem;
    align-items: center;
}

.nav-menu a {
    color: var(--white);
```

```
text-decoration: none;
font-weight: 500;
transition: opacity 0.3s;
position: relative;
}

.nav-menu a:hover {
  opacity: 0.8;
}

.cart-badge {
  background: var(--white);
  color: var(--primary-color);
  border-radius: 50%;
  padding: 2px 6px;
  font-size: 0.7rem;
  margin-left: 5px;
}

/* Main Content */
.main-content {
  flex: 1;
  padding: 2rem 0;
}

/* Flash Messages */
.flash-messages {
  position: fixed;
  top: 80px;
  right: 20px;
  z-index: 1001;
  max-width: 400px;
}

.flash-message {
  padding: 1rem;
  margin-bottom: 0.5rem;
  border-radius: 5px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  box-shadow: 0 2px 10px rgba(0,0,0,0.2);
  animation: slideIn 0.3s ease;
}

@keyframes slideIn {
  from {
    transform: translateX(400px);
  }
}
```

```
        opacity: 0;
    }
    to {
        transform: translateX(0);
        opacity: 1;
    }
}

.flash-success {
    background: var(--success-color);
    color: var(--white);
}

.flash-error {
    background: var(--danger-color);
    color: var(--white);
}

.flash-info {
    background: var(--info-color);
    color: var(--white);
}

.close-btn {
    background: none;
    border: none;
    color: var(--white);
    font-size: 1.5rem;
    cursor: pointer;
    margin-left: 1rem;
}

/* Hero Section */
.hero-section {
    background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
    color: var(--white);
    padding: 4rem 0;
    text-align: center;
    margin-bottom: 3rem;
}

.hero-section h1 {
    font-size: 3rem;
    margin-bottom: 1rem;
}

.hero-section p {
```

```
    font-size: 1.2rem;
}

/* Buttons */
.btn {
    display: inline-block;
    padding: 0.75rem 1.5rem;
    border: none;
    border-radius: 5px;
    text-decoration: none;
    cursor: pointer;
    font-size: 1rem;
    transition: all 0.3s;
    font-weight: 500;
}

.btn-primary {
    background: var(--primary-color);
    color: var(--white);
}

.btn-primary:hover {
    background: #e55a2a;
    transform: translateY(-2px);
    box-shadow: 0 4px 10px rgba(0,0,0,0.2);
}

.btn-secondary {
    background: var(--gray);
    color: var(--white);
}

.btn-secondary:hover {
    background: var(--dark-gray);
}

.btn-success {
    background: var(--success-color);
    color: var(--white);
}

.btn-danger {
    background: var(--danger-color);
    color: var(--white);
}

.btn-sm {
    padding: 0.5rem 1rem;
```

```
    font-size: 0.9rem;
}

.btn-block {
  display: block;
  width: 100%;
  text-align: center;
}

/* Restaurant Grid */
.restaurant-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 2rem;
  margin-top: 2rem;
}

.restaurant-card {
  background: var(--white);
  border-radius: 10px;
  overflow: hidden;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  transition: transform 0.3s, box-shadow 0.3s;
}

.restaurant-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 5px 20px rgba(0,0,0,0.2);
}

.restaurant-image img {
  width: 100%;
  height: 200px;
  object-fit: cover;
}

.restaurant-info {
  padding: 1.5rem;
}

.restaurant-info h3 {
  margin-bottom: 0.5rem;
  color: var(--dark-color);
}

.restaurant-meta {
  margin: 1rem 0;
  color: var(--gray);
```

```
    font-size: 0.9rem;
}

.restaurant-meta p {
    margin: 0.3rem 0;
}

/* Restaurant Detail */
.restaurant-header {
    background: var(--white);
    padding: 2rem 0;
    margin-bottom: 2rem;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.restaurant-header-content {
    display: flex;
    gap: 2rem;
    align-items: center;
}

.restaurant-header-image {
    width: 300px;
    height: 200px;
    object-fit: cover;
    border-radius: 10px;
}

.restaurant-header-info h1 {
    color: var(--dark-color);
    margin-bottom: 1rem;
}

/* Category Filter */
.category-filter {
    display: flex;
    gap: 1rem;
    margin-bottom: 2rem;
    flex-wrap: wrap;
}

.category-btn {
    padding: 0.5rem 1.5rem;
    border: 2px solid var(--primary-color);
    background: var(--white);
    color: var(--primary-color);
    border-radius: 25px;
    cursor: pointer;
}
```

```
        transition: all 0.3s;
    }

.category-btn:hover,
.category-btn.active {
    background: var(--primary-color);
    color: var(--white);
}

/* Menu Grid */
.menu-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
    gap: 2rem;
    margin-top: 2rem;
}

.menu-item-card {
    background: var(--white);
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.menu-item-card:hover {
    transform: translateY(-5px);
}

.menu-item-image img {
    width: 100%;
    height: 180px;
    object-fit: cover;
}

.menu-item-info {
    padding: 1.5rem;
}

.menu-item-info h3 {
    margin-bottom: 0.5rem;
    color: var(--dark-color);
}

.menu-item-footer {
    display: flex;
    justify-content: space-between;
    align-items: center;
}
```

```
    margin-top: 1rem;
}

.price {
    font-size: 1.3rem;
    font-weight: bold;
    color: var(--primary-color);
}

/* Cart */
.cart-container {
    display: grid;
    grid-template-columns: 2fr 1fr;
    gap: 2rem;
    margin-top: 2rem;
}

.cart-items {
    display: flex;
    flex-direction: column;
    gap: 1rem;
}

.cart-item {
    background: var(--white);
    padding: 1.5rem;
    border-radius: 10px;
    display: grid;
    grid-template-columns: 150px 1fr auto auto;
    gap: 1.5rem;
    align-items: center;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.cart-item-image {
    width: 150px;
    height: 100px;
    object-fit: cover;
    border-radius: 5px;
}

.cart-item-info h3 {
    margin-bottom: 0.5rem;
}

.cart-item-controls {
    display: flex;
    flex-direction: column;
```

```
    gap: 0.5rem;
}

.quantity-form {
  display: flex;
  align-items: center;
  gap: 0.5rem;
}

.quantity-input {
  width: 60px;
  padding: 0.5rem;
  border: 1px solid #ddd;
  border-radius: 5px;
}

.cart-item-total {
  font-size: 1.2rem;
  font-weight: bold;
  color: var(--primary-color);
}

.cart-summary {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  height: fit-content;
  position: sticky;
  top: 100px;
}

.summary-row {
  display: flex;
  justify-content: space-between;
  padding: 0.5rem 0;
  border-bottom: 1px solid #eee;
}

.summary-row.total {
  border-top: 2px solid var(--primary-color);
  margin-top: 1rem;
  padding-top: 1rem;
  font-size: 1.2rem;
  font-weight: bold;
}

/* Checkout */
```

```
.checkout-container {
  display: grid;
  grid-template-columns: 2fr 1fr;
  gap: 2rem;
  margin-top: 2rem;
}

.checkout-form-section,
.checkout-summary {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

/* Forms */
.auth-container {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 60vh;
  padding: 2rem 0;
}

.auth-card {
  background: var(--white);
  padding: 3rem;
  border-radius: 10px;
  box-shadow: 0 5px 20px rgba(0,0,0,0.1);
  width: 100%;
  max-width: 400px;
}

.auth-card h2 {
  margin-bottom: 2rem;
  text-align: center;
  color: var(--dark-color);
}

.form-container {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  max-width: 600px;
  margin: 0 auto;
}
```

```
.form-group {
  margin-bottom: 1.5rem;
}

.form-label {
  display: block;
  margin-bottom: 0.5rem;
  font-weight: 500;
  color: var(--dark-color);
}

.form-control {
  width: 100%;
  padding: 0.75rem;
  border: 1px solid #ddd;
  border-radius: 5px;
  font-size: 1rem;
  transition: border-color 0.3s;
}

.form-control:focus {
  outline: none;
  border-color: var(--primary-color);
}

.error-message {
  color: var(--danger-color);
  font-size: 0.9rem;
  margin-top: 0.5rem;
}

.auth-link {
  text-align: center;
  margin-top: 1.5rem;
  color: var(--gray);
}

.auth-link a {
  color: var(--primary-color);
  text-decoration: none;
}

.form-actions {
  display: flex;
  gap: 1rem;
  margin-top: 2rem;
}
```

```
/* Orders */
.orders-list {
  display: flex;
  flex-direction: column;
  gap: 1rem;
  margin-top: 2rem;
}

.order-card {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.order-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 1rem;
  padding-bottom: 1rem;
  border-bottom: 1px solid #eee;
}

.order-date {
  color: var(--gray);
  font-size: 0.9rem;
}

.order-status {
  padding: 0.5rem 1rem;
  border-radius: 20px;
  font-size: 0.9rem;
  font-weight: 500;
}

.status-pending {
  background: #fff3cd;
  color: #856404;
}

.status-confirmed {
  background: #d1ecf1;
  color: #0c5460;
}

.status-preparing {
  background: #d4edda;
```

```
        color: #155724;
    }

.status-out_for_delivery {
    background: #cce5ff;
    color: #004085;
}

.status-delivered {
    background: #d4edda;
    color: #155724;
}

.status-cancelled {
    background: #f8d7da;
    color: #721c24;
}

.order-details {
    margin: 1rem 0;
}

.order-detail-container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 2rem;
    margin-top: 2rem;
}

.order-info-card,
.order-items-card {
    background: var(--white);
    padding: 2rem;
    border-radius: 10px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.info-row {
    display: flex;
    justify-content: space-between;
    padding: 0.75rem 0;
    border-bottom: 1px solid #eee;
}

.order-items-list {
    margin-bottom: 1rem;
}
```

```
.order-item-detail {
  display: flex;
  justify-content: space-between;
  padding: 1rem 0;
  border-bottom: 1px solid #eee;
}

/* Admin */
.admin-nav {
  display: flex;
  gap: 1rem;
  margin-bottom: 2rem;
  flex-wrap: wrap;
}

.admin-stats {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 1rem;
  margin-bottom: 2rem;
}

.stat-card {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  text-align: center;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.stat-card h3 {
  font-size: 2.5rem;
  color: var(--primary-color);
  margin-bottom: 0.5rem;
}

.orders-table {
  background: var(--white);
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  overflow-x: auto;
}

table {
  width: 100%;
  border-collapse: collapse;
}
```

```
table th,
table td {
  padding: 1rem;
  text-align: left;
  border-bottom: 1px solid #eee;
}

table th {
  background: var(--light-color);
  font-weight: 600;
}

.status-select {
  padding: 0.5rem;
  border: 1px solid #ddd;
  border-radius: 5px;
  cursor: pointer;
}

.restaurants-admin-list,
.menu-admin-list {
  display: flex;
  flex-direction: column;
  gap: 1rem;
  margin-top: 2rem;
}

.restaurant-admin-card,
.menu-item-admin-card {
  background: var(--white);
  padding: 1.5rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  display: grid;
  grid-template-columns: 200px 1fr auto;
  gap: 2rem;
  align-items: center;
}

.restaurant-admin-card img,
.menu-item-admin-card img {
  width: 200px;
  height: 150px;
  object-fit: cover;
  border-radius: 5px;
}
```

```
.admin-actions {
    margin-bottom: 2rem;
}

/* Empty State */
.empty-state {
    text-align: center;
    padding: 4rem 2rem;
    color: var(--gray);
}

.empty-state i {
    font-size: 4rem;
    margin-bottom: 1rem;
    color: var(--gray);
}

.empty-state p {
    font-size: 1.2rem;
    margin-bottom: 1rem;
}

/* Page Title */
.page-title {
    font-size: 2.5rem;
    margin-bottom: 2rem;
    color: var(--dark-color);
}

.section-title {
    font-size: 2rem;
    margin-bottom: 1rem;
    color: var(--dark-color);
}

/* Footer */
.footer {
    background: var(--dark-color);
    color: var(--white);
    text-align: center;
    padding: 2rem 0;
    margin-top: auto;
}

/* Responsive */
@media (max-width: 768px) {
    .nav-menu {
        flex-direction: column;
    }
}
```

```

        gap: 1rem;
    }

.restaurant-grid,
.menu-grid {
    grid-template-columns: 1fr;
}

.cart-container,
.checkout-container,
.order-detail-container {
    grid-template-columns: 1fr;
}

.restaurant-header-content {
    flex-direction: column;
}

.restaurant-header-image {
    width: 100%;
}

.cart-item {
    grid-template-columns: 1fr;
}

.restaurant-admin-card,
.menu-item-admin-card {
    grid-template-columns: 1fr;
}

.hero-section h1 {
    font-size: 2rem;
}
}
}

```

Projects-J_Component\Flask\static\js\main.js:-

```

// Auto-hide flash messages after 5 seconds
document.addEventListener('DOMContentLoaded', function() {
    const flashMessages = document.querySelectorAll('.flash-message');
    flashMessages.forEach(function(message) {
        setTimeout(function() {
            message.style.animation = 'slideOut 0.3s ease';
            setTimeout(function() {
                message.remove();
            }
        })
    })
})

```

```

        }, 300);
    }, 5000);
});
});

// Add slideOut animation
const style = document.createElement('style');
style.textContent = `
@keyframes slideOut {
    from {
        transform: translateX(0);
        opacity: 1;
    }
    to {
        transform: translateX(400px);
        opacity: 0;
    }
}
`;
document.head.appendChild(style);

// Form validation
document.addEventListener('DOMContentLoaded', function() {
    const forms = document.querySelectorAll('form');
    forms.forEach(function(form) {
        form.addEventListener('submit', function(e) {
            const requiredFields = form.querySelectorAll('[required]');
            let isValid = true;

            requiredFields.forEach(function(field) {
                if (!field.value.trim()) {
                    isValid = false;
                    field.classList.add('error');
                } else {
                    field.classList.remove('error');
                }
            });

            if (!isValid) {
                e.preventDefault();
                alert('Please fill in all required fields');
            }
        });
    });
});

// Quantity input validation
document.addEventListener('DOMContentLoaded', function() {

```

```

const quantityInputs = document.querySelectorAll('.quantity-input');
quantityInputs.forEach(function(input) {
    input.addEventListener('change', function() {
        if (this.value < 1) {
            this.value = 1;
        }
    });
});

```

Projects-J_Component\Flask\templates\admin\add_menu_item.html

```

{% extends "base.html" %}

{% block title %}Add Menu Item - Admin{% endblock %}

{% block content %}


# Add Menu Item



    <div class="form-container">
        <form method="POST" action="{{ url_for('admin_add_menu_item') }}">
            {{ form.hidden_tag() }}
            <div class="form-group">
                {{ form.restaurant_id.label(class="form-label") }}
                {{ form.restaurant_id(class="form-control") }}
                {% if form.restaurant_id.errors %}
                    <div class="error-message">{{ form.restaurant_id.errors[0] }}</div>
                {% endif %}
            </div>
            <div class="form-group">
                {{ form.name.label(class="form-label") }}
                {{ form.name(class="form-control") }}
                {% if form.name.errors %}
                    <div class="error-message">{{ form.name.errors[0] }}</div>
                {% endif %}
            </div>
            <div class="form-group">
                {{ form.description.label(class="form-label") }}
                {{ form.description(class="form-control", rows=4) }}
                {% if form.description.errors %}
                    <div class="error-message">{{ form.description.errors[0] }}</div>
                {% endif %}
            </div>
        </form>
    </div>


```

```

<div class="form-group">
    {{ form.price.label(class="form-label") }}
    {{ form.price(class="form-control", step="0.01") }}
    {% if form.price.errors %}
        <div class="error-message">{{ form.price.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-group">
    {{ form.category.label(class="form-label") }}
    {{ form.category(class="form-control") }}
    {% if form.category.errors %}
        <div class="error-message">{{ form.category.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-group">
    {{ form.image_url.label(class="form-label") }}
    {{ form.image_url(class="form-control") }}
    {% if form.image_url.errors %}
        <div class="error-message">{{ form.image_url.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-actions">
    <button type="submit" class="btn btn-primary">Add Menu Item</button>
    <a href="{{ url_for('admin_restaurants') }}" class="btn btn-secondary">Cancel</a>
</div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\admin\add_restaurant.html

```

{% extends "base.html" %}

{% block title %}Add Restaurant - Admin{% endblock %}

{% block content %}
<div class="container">
    <h1 class="page-title">Add Restaurant</h1>

    <div class="form-container">

```

```

<form method="POST" action="{{ url_for('admin_add_restaurant') }}">
    {{ form.hidden_tag() }}
    <div class="form-group">
        {{ form.name.label(class="form-label") }}
        {{ form.name(class="form-control") }}
        {% if form.name.errors %}
            <div class="error-message">{{ form.name.errors[0] }}</div>
        {% endif %}
    </div>
    <div class="form-group">
        {{ form.description.label(class="form-label") }}
        {{ form.description(class="form-control", rows=4) }}
        {% if form.description.errors %}
            <div class="error-message">{{ form.description.errors[0] }}</div>
        {% endif %}
    </div>
    <div class="form-group">
        {{ form.address.label(class="form-label") }}
        {{ form.address(class="form-control") }}
        {% if form.address.errors %}
            <div class="error-message">{{ form.address.errors[0] }}</div>
        {% endif %}
    </div>
    <div class="form-group">
        {{ form.phone.label(class="form-label") }}
        {{ form.phone(class="form-control") }}
        {% if form.phone.errors %}
            <div class="error-message">{{ form.phone.errors[0] }}</div>
        {% endif %}
    </div>
    <div class="form-group">
        {{ form.image_url.label(class="form-label") }}
        {{ form.image_url(class="form-control") }}
        {% if form.image_url.errors %}
            <div class="error-message">{{ form.image_url.errors[0] }}</div>
        {% endif %}
    </div>
    <div class="form-actions">
        <button type="submit" class="btn btn-primary">Add
        Restaurant</button>
        <a href="{{ url_for('admin_restaurants') }}" class="btn btn-secondary">Cancel</a>
    </div>
</form>

```

```
    </div>
</div>
{% endblock %}
```

Projects-J_Component\Flask\templates\admin\dashboard.html

```
{% extends "base.html" %}

{% block title %}Admin Dashboard - Food Order App{% endblock %}

{% block content %}


<h1 class="page-title">Admin Dashboard</h1>

    <div class="admin-nav">
        <a href="{{ url_for('admin_restaurants') }}" class="btn btn-primary">Manage Restaurants</a>
        <a href="{{ url_for('admin_orders') }}" class="btn btn-primary">Manage Orders</a>
        <a href="{{ url_for('admin_add_restaurant') }}" class="btn btn-success">Add Restaurant</a>
        <a href="{{ url_for('admin_add_menu_item') }}" class="btn btn-success">Add Menu Item</a>
    </div>

    <div class="admin-stats">
        <div class="stat-card">
            <h3>{{ restaurants|length }}</h3>
            <p>Restaurants</p>
        </div>
        <div class="stat-card">
            <h3>{{ orders|length }}</h3>
            <p>Recent Orders</p>
        </div>
    </div>

    <div class="recent-orders">
        <h2>Recent Orders</h2>
        {% if orders %}
        <div class="orders-table">
            <table>
                <thead>
                    <tr>
                        <th>Order ID</th>
                        <th>User</th>
                        <th>Total</th>


```

```

        <th>Status</th>
        <th>Date</th>
        <th>Actions</th>
    </tr>
</thead>
<tbody>
    {% for order in orders %}
    <tr>
        <td>#{{ order.id }}</td>
        <td>{{ order.user.username }}</td>
        <td>${{ "%.2f" | format(order.total_amount + 2.99) }}</td>
        <td><span class="order-status status-{{ order.status }}">{{ order.status|replace('_', ' ')|title }}</span></td>
        <td>{{ order.created_at.strftime('%Y-%m-%d %H:%M') }}</td>
        <td><a href="{{ url_for('order_detail', order_id=order.id) }}" class="btn btn-sm">View</a></td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
{% else %}
<p>No orders yet.</p>
{% endif %}
</div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\admin\menu.html

```

{% extends "base.html" %}

{% block title %}Manage Menu - {{ restaurant.name }}{% endblock %}

{% block content %}


<h1 class="page-title">Menu - {{ restaurant.name }}</h1>

    <div class="admin-actions">
        <a href="{{ url_for('admin_add_menu_item') }}" class="btn btn-success">
            <i class="fas fa-plus"></i> Add Menu Item
        </a>
    </div>


```

```

        <a href="{{ url_for('admin_restaurants') }}" class="btn btn-
secondary">Back to Restaurants</a>
    </div>

    <div class="menu-admin-list">
        {% for item in menu_items %}
        <div class="menu-item-admin-card">
            
            <div class="menu-item-admin-info">
                <h3>{{ item.name }}</h3>
                <p>{{ item.description or 'No description' }}</p>
                <p><strong>Price:</strong> ${{ "%2f" | format(item.price)
}}</p>
                <p><strong>Category:</strong> {{ item.category or
'Uncategorized' }}</p>
                <p><strong>Status:</strong>
                    <span class="status-{{ 'available' if item.is_available
else 'unavailable' }}">
                        {{ 'Available' if item.is_available else 'Unavailable'
}}
                    </span>
                </p>
            </div>
        </div>
        {% else %}
        <div class="empty-state">
            <p>No menu items found. <a href="{{ url_for('admin_add_menu_item')
}}>Add one now</a></p>
        </div>
        {% endfor %}
    </div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\admin\orders.html

```

{% extends "base.html" %}

{% block title %}Manage Orders - Admin{% endblock %}

{% block content %}
<div class="container">
    <h1 class="page-title">Manage Orders</h1>
    <div class="orders-table">

```

```

<table>
    <thead>
        <tr>
            <th>Order ID</th>
            <th>User</th>
            <th>Total</th>
            <th>Status</th>
            <th>Date</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tbody>
        {% for order in orders %}
        <tr>
            <td>{{ order.id }}</td>
            <td>{{ order.user.username }}</td>
            <td>${{ "%.2f" | format(order.total_amount + 2.99) }}</td>
            <td>
                <form method="POST" action="{{ url_for('admin_update_order_status', order_id=order.id) }}" style="display: inline;">
                    <select name="status" onchange="this.form.submit()" class="status-select">
                        <option value="pending" {% if order.status == 'pending' %}selected{% endif %}>Pending</option>
                        <option value="confirmed" {% if order.status == 'confirmed' %}selected{% endif %}>Confirmed</option>
                        <option value="preparing" {% if order.status == 'preparing' %}selected{% endif %}>Preparing</option>
                        <option value="out_for_delivery" {% if order.status == 'out_for_delivery' %}selected{% endif %}>Out for Delivery</option>
                        <option value="delivered" {% if order.status == 'delivered' %}selected{% endif %}>Delivered</option>
                        <option value="cancelled" {% if order.status == 'cancelled' %}selected{% endif %}>Cancelled</option>
                    </select>
                </form>
            </td>
            <td>{{ order.created_at.strftime('%Y-%m-%d %H:%M') }}</td>
            <td><a href="{{ url_for('order_detail', order_id=order.id) }}>View</a></td>
        </tr>
        {% else %}
        <tr>
            <td colspan="6" class="empty-state">No orders found.</td>
        </tr>
        {% endfor %}
    </tbody>

```

```

        </tbody>
    </table>
</div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\admin\restaurants.html

```

{% extends "base.html" %}

{% block title %}Manage Restaurants - Admin{% endblock %}

{% block content %}


<h1 class="page-title">Manage Restaurants</h1>

    <div class="admin-actions">
        <a href="{{ url_for('admin_add_restaurant') }}" class="btn btn-success">
            <i class="fas fa-plus"></i> Add Restaurant
        </a>
    </div>

    <div class="restaurants-admin-list">
        {% for restaurant in restaurants %}
        <div class="restaurant-admin-card">
            
            <div class="restaurant-admin-info">
                <h3>{{ restaurant.name }}</h3>
                <p>{{ restaurant.description or 'No description' }}</p>
                <p><strong>Status:</strong>
                    <span class="status-{{ 'active' if restaurant.is_active
else 'inactive' }}">
                        {{ 'Active' if restaurant.is_active else 'Inactive' }}
                    </span>
                </p>
            </div>
            <div class="restaurant-admin-actions">
                <a href="{{ url_for('admin_restaurant_menu',
restaurant_id=restaurant.id) }}" class="btn btn-primary">
                    Manage Menu
                </a>
            </div>
        </div>
        {% endfor %}
    </div>


```

```

    {% else %}
        <div class="empty-state">
            <p>No restaurants found. <a href="{{ url_for('admin_add_restaurant') }}">Add one now</a></p>
        </div>
    {% endfor %}
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Food Order App{% endblock %}</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
</head>
<body>
    <nav class="navbar">
        <div class="container">
            <a href="{{ url_for('index') }}" class="logo">
                <i class="fas fa-utensils"></i> FoodOrder
            </a>
            <ul class="nav-menu">
                <li><a href="{{ url_for('index') }}">Home</a></li>
                {% if current_user.is_authenticated %}
                    <li><a href="{{ url_for('cart') }}">
                        <i class="fas fa-shopping-cart"></i> Cart
                        {% if current_user.cart_items|length > 0 %}
                            <span class="cart-badge">{{ current_user.cart_items|length }}</span>
                        {% endif %}
                    </a></li>
                    <li><a href="{{ url_for('order_history') }}">My Orders</a></li>
                    {% if current_user.is_admin %}
                        <li><a href="{{ url_for('admin_dashboard') }}">Admin</a></li>
                    {% endif %}
                {% endif %}
            </ul>
        </div>
    </nav>

```

```

        <li><a href="{{ url_for('logout') }}">Logout ({{ current_user.username }})</a></li>
        {% else %}
            <li><a href="{{ url_for('login') }}">Login</a></li>
            <li><a href="{{ url_for('register') }}">Register</a></li>
        {% endif %}
    </ul>
</div>
</nav>

<main class="main-content">
    {% with messages = get_flashed_messages(with_categories=true) %}
        {% if messages %}
            <div class="flash-messages">
                {% for category, message in messages %}
                    <div class="flash-message flash-{{ category }}">
                        <span>{{ message }}</span>
                        <button class="close-btn"
onclick="this.parentElement.remove()">&times;</button>
                    </div>
                {% endfor %}
            </div>
        {% endif %}
    {% endwith %}

    {% block content %}{% endblock %}
</main>

<footer class="footer">
    <div class="container">
        <p>&copy; 2025 FoodOrder App. All rights reserved.</p>
    </div>
</footer>

<script src="{{ url_for('static', filename='js/main.js') }}"></script>
    {% block scripts %}{% endblock %}
</body>
</html>

```

Projects-J_Component\Flask\templates\cart.html

```

{% extends "base.html" %}

{% block title %}Shopping Cart - Food Order App{% endblock %}

{% block content %}

```

```

<div class="container">
    <h1 class="page-title">Shopping Cart</h1>

    {% if cart_items %}
        <div class="cart-container">
            <div class="cart-items">
                {% for item in cart_items %}
                    <div class="cart-item">
                        
                        <div class="cart-item-info">
                            <h3>{{ item.menu_item.name }}</h3>
                            <p>{{ item.menu_item.restaurant.name }}</p>
                            <p class="cart-item-price">${{ "%2f" | format(item.menu_item.price) }} each</p>
                        </div>
                        <div class="cart-item-controls">
                            <form method="POST" action="{{ url_for('update_cart',
                                cart_item_id=item.id) }}" class="quantity-form">
                                <label>Quantity:</label>
                                <input type="number" name="quantity" value="{{
                                    item.quantity }}" min="1" class="quantity-input">
                                <button type="submit" class="btn btn-sm">Update</button>
                            </form>
                            <form method="POST" action="{{ url_for('remove_from_cart',
                                cart_item_id=item.id) }}" style="display: inline;">
                                <button type="submit" class="btn btn-danger btn-sm">
                                    <i class="fas fa-trash"></i> Remove
                                </button>
                            </form>
                        </div>
                        <div class="cart-item-total">
                            <strong>${{ "%2f" | format(item.menu_item.price *
                                item.quantity) }}</strong>
                        </div>
                    </div>
                {% endfor %}
            </div>

            <div class="cart-summary">
                <h3>Order Summary</h3>
                <div class="summary-row">
                    <span>Subtotal:</span>
                    <span>${{ "%2f" | format(total) }}</span>
                </div>
                <div class="summary-row">

```

```

        <span>Delivery Fee:</span>
        <span>$2.99</span>
    </div>
    <div class="summary-row total">
        <span>Total:</span>
        <span>${{ "%2f" | format(total + 2.99) }}</span>
    </div>
    <a href="{{ url_for('checkout') }}" class="btn btn-primary btn-block">Proceed to Checkout</a>
</div>
<% else %>
<div class="empty-state">
    <i class="fas fa-shopping-cart"></i>
    <p>Your cart is empty.</p>
    <a href="{{ url_for('index') }}" class="btn btn-primary">Browse Restaurants</a>
</div>
<% endif %>
</div>
<% endblock %>

```

Projects-J_Component\Flask\templates\checkout.html

```

{% extends "base.html" %}

{% block title %}Checkout - Food Order App{% endblock %}

{% block content %}
<div class="container">
    <h1 class="page-title">Checkout</h1>

    <div class="checkout-container">
        <div class="checkout-form-section">
            <h2>Delivery Information</h2>
            <form method="POST" action="{{ url_for('checkout') }}">
                {{ form.hidden_tag() }}
                <div class="form-group">
                    {{ form.delivery_address.label(class="form-label") }}
                    {{ form.delivery_address(class="form-control",
placeholder="Enter your delivery address") }}
                    {% if form.delivery_address.errors %}
                        <div class="error-message">{{ form.delivery_address.errors[0] }}</div>
                    {% endif %}
                </div>
            
```

```

        <div class="form-group">
            {{ form.phone.label(class="form-label") }}
            {{ form.phone(class="form-control", placeholder="Enter
your phone number") }}
            {% if form.phone.errors %}
                <div class="error-message">{{ form.phone.errors[0]
}}</div>
            {% endif %}
        </div>
        <button type="submit" class="btn btn-primary btn-block">Place
Order</button>
    </form>
</div>

<div class="checkout-summary">
    <h2>Order Summary</h2>
    <div class="order-items">
        {% for item in cart_items %}
            <div class="order-item">
                <span>{{ item.menu_item.name }} x {{ item.quantity
}}</span>
                <span>${{ "%.2f" | format(item.menu_item.price *
item.quantity) }}</span>
            </div>
        {% endfor %}
    </div>
    <div class="summary-row">
        <span>Subtotal:</span>
        <span>${{ "%.2f" | format(total) }}</span>
    </div>
    <div class="summary-row">
        <span>Delivery Fee:</span>
        <span>$2.99</span>
    </div>
    <div class="summary-row total">
        <span>Total:</span>
        <span>${{ "%.2f" | format(total + 2.99) }}</span>
    </div>
</div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\index.html

```
{% extends "base.html" %}
```

```

{% block title %}Home - Food Order App{% endblock %}

{% block content %}


# Welcome to FoodOrder



Order delicious food from your favorite restaurants



## Featured Restaurants



{% for restaurant in restaurants %}


![{{ restaurant.name }}]({{ restaurant.image_url or 'https://via.placeholder.com/400x300?text=Restaurant' }})



### {{ restaurant.name }}



{{ restaurant.description or 'Delicious food awaits!' }}



{% if restaurant.address %}


{{ restaurant.address }}


{% endif %}
{% if restaurant.phone %}


{{ restaurant.phone }}


{% endif %}


View Menu


{% else %}


No restaurants available at the moment.


{% if current_user.is_admin %}

Add Restaurant

{% endif %}


```

```

        </div>
    {% endfor %}
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\login.html

```

{% extends "base.html" %}

{% block title %}Login - Food Order App{% endblock %}

{% block content %}


<div class="auth-card">
        <h2>Login</h2>
        <form method="POST" action="{{ url_for('login') }}>
            {{ form.hidden_tag() }}
            <div class="form-group">
                {{ form.username.label(class="form-label") }}
                {{ form.username(class="form-control") }}
                {% if form.username.errors %}
                    <div class="error-message">{{ form.username.errors[0] }}</div>
                {% endif %}
            </div>
            <div class="form-group">
                {{ form.password.label(class="form-label") }}
                {{ form.password(class="form-control") }}
                {% if form.password.errors %}
                    <div class="error-message">{{ form.password.errors[0] }}</div>
                {% endif %}
            </div>
            <button type="submit" class="btn btn-primary btn-block">Login</button>
        </form>
        <p class="auth-link">Don't have an account? <a href="{{ url_for('register') }}>Register here</a></p>
    </div>
</div>
{% endblock %}


```

Projects-J_Component\Flask\templates\order_detail.html

```

{% extends "base.html" %}

{% block title %}Order #{{ order.id }} - Food Order App{% endblock %}

{% block content %}


# Order #{{ order.id }}



## Order Information



Order Date:
                {{ order.created_at.strftime('%B %d, %Y at %I:%M %p') }}



Status:
                
                    {{ order.status|replace('_', ' ')|title }}



Total Amount:
${{ "%.2f" | format(order.total_amount + 2.99) }}


            {% if order.delivery_address %}
            

Delivery Address:
{{ order.delivery_address }}


            {% endif %}
            {% if order.phone %}
            

Phone:
{{ order.phone }}


            {% endif %}



## Order Items



{% for item in order.order_items %}
                

#### {{ item.menu_item.name }}


```

```

        <p>Quantity: {{ item.quantity }}</p>
        <p>Price: ${{ "%.2f" | format(item.price) }} each</p>
    </div>
    <div class="order-item-total">
        ${{ "%.2f" | format(item.price * item.quantity) }}
    </div>
</div>
{% endfor %}
</div>
<div class="order-summary">
    <div class="summary-row">
        <span>Subtotal:</span>
        <span>${{ "%.2f" | format(order.total_amount) }}</span>
    </div>
    <div class="summary-row">
        <span>Delivery Fee:</span>
        <span>$2.99</span>
    </div>
    <div class="summary-row total">
        <span>Total:</span>
        <span>${{ "%.2f" | format(order.total_amount + 2.99) }}</span>
    </div>
</div>
</div>
<a href="{{ url_for('order_history') }}" class="btn btn-secondary">Back to Orders</a>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\orders.html

```

{% extends "base.html" %}

{% block title %}My Orders - Food Order App{% endblock %}

{% block content %}
<div class="container">
    <h1 class="page-title">My Orders</h1>

    {% if orders %}
    <div class="orders-list">
        {% for order in orders %}
        <div class="order-card">

```

```

<div class="order-header">
    <div>
        <h3>Order #{{ order.id }}</h3>
        <p class="order-date">{{ order.created_at.strftime('%B %Y at %I:%M %p') }}</p>
    </div>
    <div class="order-status status-{{ order.status }}">
        {{ order.status|replace('_', ' ')|title }}
    </div>
</div>
<div class="order-details">
    <p><strong>Total:</strong> ${{
    "%2f" | format(order.total_amount + 2.99) }}</p>
    <p><strong>Items:</strong> {{ order.total_items }} item(s)</p>
    {% if order.delivery_address %}
        <p><strong>Delivery Address:</strong> {{ order.delivery_address }}</p>
        {% endif %}
    </div>
    <a href="{{ url_for('order_detail', order_id=order.id) }}" class="btn btn-primary">View Details</a>
</div>
{% endfor %}
</div>
{% else %}
<div class="empty-state">
    <i class="fas fa-receipt"></i>
    <p>You haven't placed any orders yet.</p>
    <a href="{{ url_for('index') }}" class="btn btn-primary">Browse Restaurants</a>
</div>
{% endif %}
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\register.html

```

{% extends "base.html" %}

{% block title %}Register - Food Order App{% endblock %}

{% block content %}
<div class="auth-container">
    <div class="auth-card">
        <h2>Register</h2>
        <form method="POST" action="{{ url_for('register') }}">

```

```

{{ form.hidden_tag() }}
<div class="form-group">
    {{ form.username.label(class="form-label") }}
    {{ form.username(class="form-control") }}
    {% if form.username.errors %}
        <div class="error-message">{{ form.username.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-group">
    {{ form.email.label(class="form-label") }}
    {{ form.email(class="form-control") }}
    {% if form.email.errors %}
        <div class="error-message">{{ form.email.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-group">
    {{ form.password.label(class="form-label") }}
    {{ form.password(class="form-control") }}
    {% if form.password.errors %}
        <div class="error-message">{{ form.password.errors[0] }}</div>
    {% endif %}
</div>
<div class="form-group">
    {{ form.confirm_password.label(class="form-label") }}
    {{ form.confirm_password(class="form-control") }}
    {% if form.confirm_password.errors %}
        <div class="error-message">{{ form.confirm_password.errors[0] }}</div>
    {% endif %}
</div>
<button type="submit" class="btn btn-primary btn-block">Register</button>
</form>
<p class="auth-link">Already have an account? <a href="{{ url_for('login') }}>Login here</a></p>
</div>
</div>
{% endblock %}

```

Projects-J_Component\Flask\templates\restaurant.html

```
{% extends "base.html" %}
```

```

{% block title %}{{ restaurant.name }} - Food Order App{% endblock %}

{% block content %}


![{{ restaurant.name }}]({{ restaurant.image_url or 'https://via.placeholder.com/400x300?text=Restaurant' }})

# {{ restaurant.name }}



{{ restaurant.description or 'Delicious food awaits!' }}



All



{{ category }}



![{{ item.name }}]({{ item.image_url or 'https://via.placeholder.com/300x200?text=Food' }})



### {{ item.name }}



{{ item.description or 'Delicious food item' }}


```

```

<span class="price">${{ "%%.2f" | format(item.price)
}}</span>
        {% if current_user.is_authenticated %}
            <form method="POST" action="{{ url_for('add_to_cart',
item_id=item.id) }}" style="display: inline;">
                <button type="submit" class="btn btn-primary btn-sm">
                    <i class="fas fa-cart-plus"></i> Add to Cart
                </button>
            </form>
        {% else %}
            <a href="{{ url_for('login') }}" class="btn btn-primary
btn-sm">Login to Order</a>
        {% endif %}
    </div>
</div>
{% else %}
    <div class="empty-state">
        <i class="fas fa-utensils"></i>
        <p>No menu items available.</p>
    </div>
    {% endfor %}
</div>
</div>
{% endblock %}

{% block scripts %}
<script>
    document.querySelectorAll('.category-btn').forEach(btn => {
        btn.addEventListener('click', function() {
            document.querySelectorAll('.category-btn').forEach(b =>
b.classList.remove('active'));
            this.classList.add('active');

            const category = this.dataset.category;
            document.querySelectorAll('.menu-item-card').forEach(card => {
                if (category === 'all' || card.dataset.category === category)
{
                    card.style.display = 'block';
                } else {
                    card.style.display = 'none';
                }
            });
        });
    });
</script>
{% endblock %}

```

Projects-J_Component\Flask\app.py

```
from flask import Flask, render_template, request, redirect, url_for, flash, jsonify
from flask_login import LoginManager, login_user, logout_user, login_required, current_user
from models import db, User, Restaurant, MenuItem, CartItem, Order, OrderItem
from forms import LoginForm, RegisterForm, RestaurantForm, MenuItemForm, OrderForm
from config import Config
from datetime import datetime
import time
import os

app = Flask(__name__)
app.config.from_object(Config)

db.init_app(app)
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'
login_manager.login_message = 'Please log in to access this page.'

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))

# Health check endpoint
@app.route('/health')
def health_check():
    """Health check endpoint for Docker and load balancers"""
    try:
        # Check database connection
        db.session.execute(db.text('SELECT 1'))
        db_status = 'healthy'
    except Exception as e:
        db_status = f'unhealthy: {str(e)}'

    return jsonify({
        'status': 'healthy' if db_status == 'healthy' else 'degraded',
        'database': db_status,
        'timestamp': datetime.utcnow().isoformat()
    }), 200 if db_status == 'healthy' else 503

# Initialize database with retry logic for containerized environments
def init_db():
    """Initialize database with retry logic for container startup"""


```

```

max_retries = 30
retry_delay = 2

for attempt in range(max_retries):
    try:
        with app.app_context():
            db.create_all()
            # Create admin user if it doesn't exist
            if not User.query.filter_by(username='admin').first():
                admin = User(username='admin', email='admin@foodapp.com')
                admin.set_password('admin123')
                admin.is_admin = True
                db.session.add(admin)
                db.session.commit()
                print("Database initialized successfully")
            return True
    except Exception as e:
        if attempt < max_retries - 1:
            print(f"Database initialization attempt {attempt + 1} failed: {e}. Retrying in {retry_delay} seconds...")
            time.sleep(retry_delay)
        else:
            print(f"Failed to initialize database after {max_retries} attempts: {e}")
            raise
    return False

# Initialize database on startup
if os.environ.get('INIT_DB', 'true').lower() == 'true':
    init_db()

# Home page
@app.route('/')
def index():
    restaurants = Restaurant.query.filter_by(is_active=True).all()
    return render_template('index.html', restaurants=restaurants)

# Authentication routes
@app.route('/login', methods=['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if user and user.check_password(form.password.data):
            login_user(user)

```

```

        flash('Login successful!', 'success')
        next_page = request.args.get('next')
        return redirect(next_page) if next_page else
redirect(url_for('index'))
    else:
        flash('Invalid username or password', 'error')
    return render_template('login.html', form=form)

@app.route('/register', methods=['GET', 'POST'])
def register():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = RegisterForm()
    if form.validate_on_submit():
        if User.query.filter_by(username=form.username.data).first():
            flash('Username already exists', 'error')
        elif User.query.filter_by(email=form.email.data).first():
            flash('Email already registered', 'error')
        else:
            user = User(username=form.username.data, email=form.email.data)
            user.set_password(form.password.data)
            db.session.add(user)
            db.session.commit()
            flash('Registration successful! Please log in.', 'success')
            return redirect(url_for('login'))
    return render_template('register.html', form=form)

@app.route('/logout')
@login_required
def logout():
    logout_user()
    flash('You have been logged out', 'info')
    return redirect(url_for('index'))

# Restaurant and menu routes
@app.route('/restaurant/<int:restaurant_id>')
def restaurant_detail(restaurant_id):
    restaurant = Restaurant.query.get_or_404(restaurant_id)
    menu_items = MenuItem.query.filter_by(restaurant_id=restaurant_id,
is_available=True).all()
    categories =
db.session.query(MenuItem.category).filter_by(restaurant_id=restaurant_id).dis-
tinct().all()
    categories = [cat[0] for cat in categories if cat[0]]
    return render_template('restaurant.html', restaurant=restaurant,
menu_items=menu_items, categories=categories)

```

```

# Cart routes
@app.route('/cart')
@login_required
def cart():
    cart_items = CartItem.query.filter_by(user_id=current_user.id).all()
    total = sum(item.menu_item.price * item.quantity for item in cart_items)
    return render_template('cart.html', cart_items=cart_items, total=total)

@app.route('/add_to_cart/<int:item_id>', methods=['POST'])
@login_required
def add_to_cart(item_id):
    menu_item = MenuItem.query.get_or_404(item_id)
    if not menu_item.is_available:
        flash('Item is not available', 'error')
        return redirect(url_for('restaurant_detail',
                                restaurant_id=menu_item.restaurant_id))

    cart_item = CartItem.query.filter_by(user_id=current_user.id,
                                         menu_item_id=item_id).first()
    if cart_item:
        cart_item.quantity += 1
    else:
        cart_item = CartItem(user_id=current_user.id, menu_item_id=item_id,
                             quantity=1)
        db.session.add(cart_item)

    db.session.commit()
    flash(f'{menu_item.name} added to cart!', 'success')
    return redirect(url_for('restaurant_detail',
                            restaurant_id=menu_item.restaurant_id))

@app.route('/update_cart/<int:cart_item_id>', methods=['POST'])
@login_required
def update_cart(cart_item_id):
    cart_item = CartItem.query.get_or_404(cart_item_id)
    if cart_item.user_id != current_user.id:
        flash('Unauthorized', 'error')
        return redirect(url_for('cart'))

    quantity = int(request.form.get('quantity', 1))
    if quantity <= 0:
        db.session.delete(cart_item)
    else:
        cart_item.quantity = quantity
    db.session.commit()
    return redirect(url_for('cart'))

@app.route('/remove_from_cart/<int:cart_item_id>', methods=['POST'])

```

```

@login_required
def remove_from_cart(cart_item_id):
    cart_item = CartItem.query.get_or_404(cart_item_id)
    if cart_item.user_id != current_user.id:
        flash('Unauthorized', 'error')
        return redirect(url_for('cart'))

    db.session.delete(cart_item)
    db.session.commit()
    flash('Item removed from cart', 'info')
    return redirect(url_for('cart'))


# Order routes
@app.route('/checkout', methods=['GET', 'POST'])
@login_required
def checkout():
    cart_items = CartItem.query.filter_by(user_id=current_user.id).all()
    if not cart_items:
        flash('Your cart is empty', 'error')
        return redirect(url_for('cart'))

    form = OrderForm()
    if form.validate_on_submit():
        total = sum(item.menu_item.price * item.quantity for item in
                    cart_items)
        order = Order(
            user_id=current_user.id,
            total_amount=total,
            delivery_address=form.delivery_address.data,
            phone=form.phone.data,
            status='pending'
        )
        db.session.add(order)
        db.session.flush()

        for cart_item in cart_items:
            order_item = OrderItem(
                order_id=order.id,
                menu_item_id=cart_item.menu_item_id,
                quantity=cart_item.quantity,
                price=cart_item.menu_item.price
            )
            db.session.add(order_item)
            db.session.delete(cart_item)

    db.session.commit()
    flash('Order placed successfully!', 'success')
    return redirect(url_for('order_history'))

```

```

        total = sum(item.menu_item.price * item.quantity for item in cart_items)
        return render_template('checkout.html', cart_items=cart_items,
total=total, form=form)

@app.route('/orders')
@login_required
def order_history():
    orders =
Order.query.filter_by(user_id=current_user.id).order_by(Order.created_at.desc()).all()
    # Calculate total items for each order
    for order in orders:
        order.total_items = sum(item.quantity for item in order.order_items)
    return render_template('orders.html', orders=orders)

@app.route('/order/<int:order_id>')
@login_required
def order_detail(order_id):
    order = Order.query.get_or_404(order_id)
    if order.user_id != current_user.id and not current_user.is_admin:
        flash('Unauthorized', 'error')
        return redirect(url_for('index'))
    return render_template('order_detail.html', order=order)

# Admin routes
@app.route('/admin')
@login_required
def admin_dashboard():
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    restaurants = Restaurant.query.all()
    orders = Order.query.order_by(Order.created_at.desc()).limit(10).all()
    return render_template('admin/dashboard.html', restaurants=restaurants,
orders=orders)

@app.route('/admin/restaurants')
@login_required
def admin_restaurants():
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    restaurants = Restaurant.query.all()
    return render_template('admin/restaurants.html', restaurants=restaurants)

```

```

@app.route('/admin/restaurant/add', methods=['GET', 'POST'])
@login_required
def admin_add_restaurant():
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    form = RestaurantForm()
    if form.validate_on_submit():
        restaurant = Restaurant(
            name=form.name.data,
            description=form.description.data,
            address=form.address.data,
            phone=form.phone.data,
            image_url=form.image_url.data or
            'https://via.placeholder.com/400x300?text=Restaurant'
        )
        db.session.add(restaurant)
        db.session.commit()
        flash('Restaurant added successfully!', 'success')
        return redirect(url_for('admin_restaurants'))
    return render_template('admin/add_restaurant.html', form=form)

@app.route('/admin/restaurant/<int:restaurant_id>/menu')
@login_required
def admin_restaurant_menu(restaurant_id):
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    restaurant = Restaurant.query.get_or_404(restaurant_id)
    menu_items = MenuItem.query.filter_by(restaurant_id=restaurant_id).all()
    return render_template('admin/menu.html', restaurant=restaurant,
    menu_items=menu_items)

@app.route('/admin/menu/add', methods=['GET', 'POST'])
@login_required
def admin_add_menu_item():
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    form = MenuItemForm()
    form.restaurant_id.choices = [(r.id, r.name) for r in
    Restaurant.query.all()]

    if form.validate_on_submit():
        menu_item = MenuItem(

```

```

        name=form.name.data,
        description=form.description.data,
        price=form.price.data,
        category=form.category.data,
        image_url=form.image_url.data or
        'https://via.placeholder.com/300x200?text=Food',
        restaurant_id=form.restaurant_id.data
    )
    db.session.add(menu_item)
    db.session.commit()
    flash('Menu item added successfully!', 'success')
    return redirect(url_for('admin_restaurant_menu',
    restaurant_id=form.restaurant_id.data))
    return render_template('admin/add_menu_item.html', form=form)

@app.route('/admin/orders')
@login_required
def admin_orders():
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    orders = Order.query.order_by(Order.created_at.desc()).all()
    return render_template('admin/orders.html', orders=orders)

@app.route('/admin/order/<int:order_id>/update_status', methods=['POST'])
@login_required
def admin_update_order_status(order_id):
    if not current_user.is_admin:
        flash('Access denied', 'error')
        return redirect(url_for('index'))

    order = Order.query.get_or_404(order_id)
    new_status = request.form.get('status')
    if new_status in ['pending', 'confirmed', 'preparing', 'out_for_delivery',
    'delivered', 'cancelled']:
        order.status = new_status
        order.updated_at = datetime.utcnow()
        db.session.commit()
        flash('Order status updated!', 'success')
    return redirect(url_for('admin_orders'))

if __name__ == '__main__':
    # Use environment variable for host/port or defaults
    host = os.environ.get('FLASK_HOST', '0.0.0.0')
    port = int(os.environ.get('FLASK_PORT', 5000))
    debug = app.config.get('FLASK_DEBUG', False)
    app.run(host=host, port=port, debug=debug)

```

Projects-J_Component\Flask\config.py

```
import os

class Config:
    SECRET_KEY = os.environ.get('SECRET_KEY') or 'dev-secret-key-change-in-production'

    # Database configuration - PostgreSQL for production, SQLite for Local dev
    DB_HOST = os.environ.get('DB_HOST', 'localhost')
    DB_PORT = os.environ.get('DB_PORT', '5432')
    DB_NAME = os.environ.get('DB_NAME', 'food_order_db')
    DB_USER = os.environ.get('DB_USER', 'foodapp_user')
    DB_PASSWORD = os.environ.get('DB_PASSWORD', 'foodapp_pass')

    # Use PostgreSQL if DB_HOST is set to a non-localhost value, otherwise
    # SQLite
    # Check if we're in a containerized environment (DB_HOST is not localhost)
    _db_host = os.environ.get('DB_HOST', 'localhost')
    if _db_host and _db_host != 'localhost':
        SQLALCHEMY_DATABASE_URI =
f"postgresql://{{DB_USER}}:{{DB_PASSWORD}}@{{DB_HOST}}:{{DB_PORT}}/{{DB_NAME}}"
    else:
        SQLALCHEMY_DATABASE_URI = os.environ.get('DATABASE_URL') or
'sqlite:///food_order.db'

    SQLALCHEMY_TRACK_MODIFICATIONS = False

    # Redis configuration
    REDIS_HOST = os.environ.get('REDIS_HOST', 'localhost')
    REDIS_PORT = int(os.environ.get('REDIS_PORT', 6379))
    REDIS_DB = int(os.environ.get('REDIS_DB', 0))

    # Flask settings
    FLASK_ENV = os.environ.get('FLASK_ENV', 'development')
    FLASK_DEBUG = os.environ.get('FLASK_DEBUG', 'True').lower() == 'true'
```

Projects-J_Component\Flask\forms.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, TextAreaField, FloatField,
IntegerField, SelectField, BooleanField
```

```

from wtforms.validators import DataRequired, Email, EqualTo, Length,
NumberRange

class LoginForm(FlaskForm):
    username = StringField('Username', validators=[DataRequired()])
    password = PasswordField('Password', validators=[DataRequired()])

class RegisterForm(FlaskForm):
    username = StringField('Username', validators=[DataRequired(),
Length(min=4, max=20)])
    email = StringField('Email', validators=[DataRequired(), Email()])
    password = PasswordField('Password', validators=[DataRequired(),
Length(min=6)])
    confirm_password = PasswordField('Confirm Password',
validators=[DataRequired(), EqualTo('password')])

class RestaurantForm(FlaskForm):
    name = StringField('Restaurant Name', validators=[DataRequired()])
    description = TextAreaField('Description')
    address = StringField('Address')
    phone = StringField('Phone')
    image_url = StringField('Image URL')

class MenuItemForm(FlaskForm):
    name = StringField('Item Name', validators=[DataRequired()])
    description = TextAreaField('Description')
    price = FloatField('Price', validators=[DataRequired(),
NumberRange(min=0)])
    category = StringField('Category')
    image_url = StringField('Image URL')
    restaurant_id = SelectField('Restaurant', coerce=int,
validators=[DataRequired()])

class OrderForm(FlaskForm):
    delivery_address = StringField('Delivery Address',
validators=[DataRequired()])
    phone = StringField('Phone Number', validators=[DataRequired()])

```

Projects-J_Component\Flask\models.py

```

from flask_sqlalchemy import SQLAlchemy
from flask_login import UserMixin
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime

db = SQLAlchemy()

```

```

class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password_hash = db.Column(db.String(255), nullable=False)
    is_admin = db.Column(db.Boolean, default=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

    # Relationships
    orders = db.relationship('Order', backref='user', lazy=True)
    cart_items = db.relationship('CartItem', backref='user', lazy=True,
                                cascade='all, delete-orphan')

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)

class Restaurant(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.Text)
    image_url = db.Column(db.String(255))
    address = db.Column(db.String(200))
    phone = db.Column(db.String(20))
    is_active = db.Column(db.Boolean, default=True)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

    # Relationships
    menu_items = db.relationship('MenuItem', backref='restaurant', lazy=True,
                                cascade='all, delete-orphan')

class MenuItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.Text)
    price = db.Column(db.Float, nullable=False)
    image_url = db.Column(db.String(255))
    category = db.Column(db.String(50))
    is_available = db.Column(db.Boolean, default=True)
    restaurant_id = db.Column(db.Integer, db.ForeignKey('restaurant.id'),
                            nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

    # Relationships

```

```

    cart_items = db.relationship('CartItem', backref='menu_item', lazy=True,
                                cascade='all, delete-orphan')
    order_items = db.relationship('OrderItem', backref='menu_item', lazy=True)

class CartItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    menu_item_id = db.Column(db.Integer, db.ForeignKey('menu_item.id'),
                           nullable=False)
    quantity = db.Column(db.Integer, default=1, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

class Order(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    total_amount = db.Column(db.Float, nullable=False)
    status = db.Column(db.String(20), default='pending') # pending,
    confirmed, preparing, out_for_delivery, delivered, cancelled
    delivery_address = db.Column(db.String(200))
    phone = db.Column(db.String(20))
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    updated_at = db.Column(db.DateTime, default=datetime.utcnow,
                           onupdate=datetime.utcnow)

    # Relationships
    order_items = db.relationship('OrderItem', backref='order', lazy=True,
                                cascade='all, delete-orphan')

class OrderItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    order_id = db.Column(db.Integer, db.ForeignKey('order.id'),
                        nullable=False)
    menu_item_id = db.Column(db.Integer, db.ForeignKey('menu_item.id'),
                           nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
    price = db.Column(db.Float, nullable=False) # Price at time of order

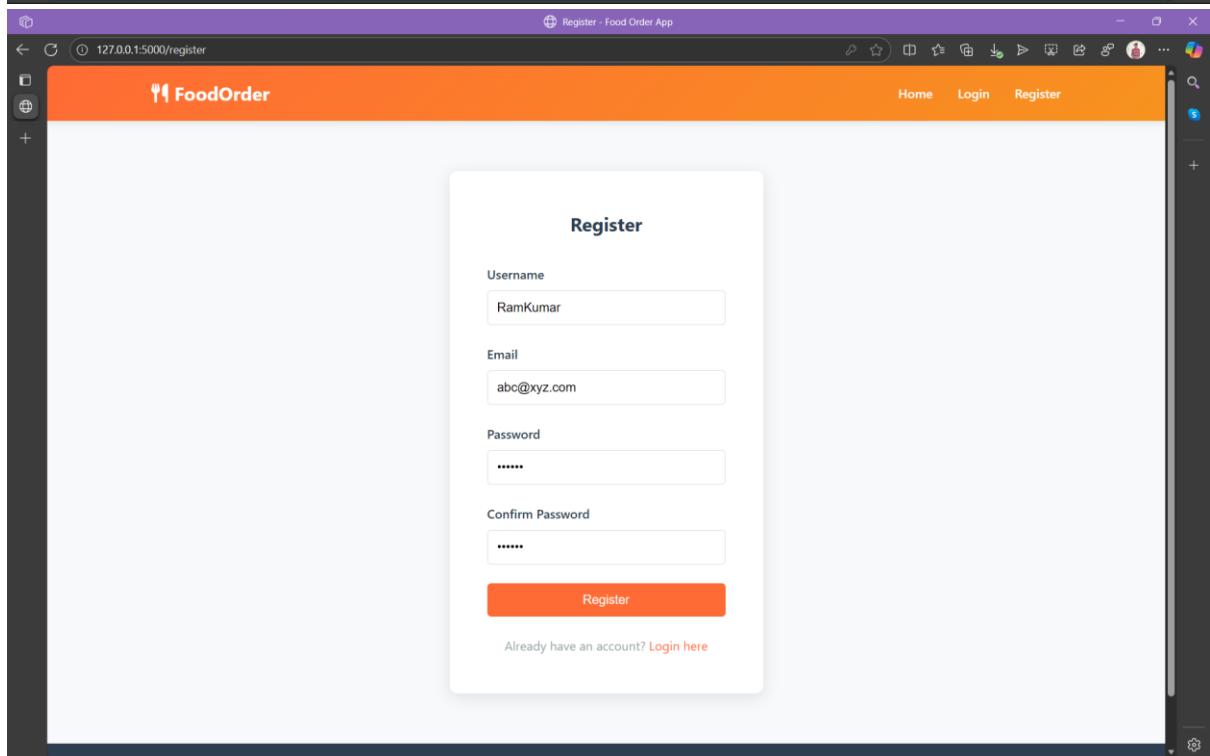
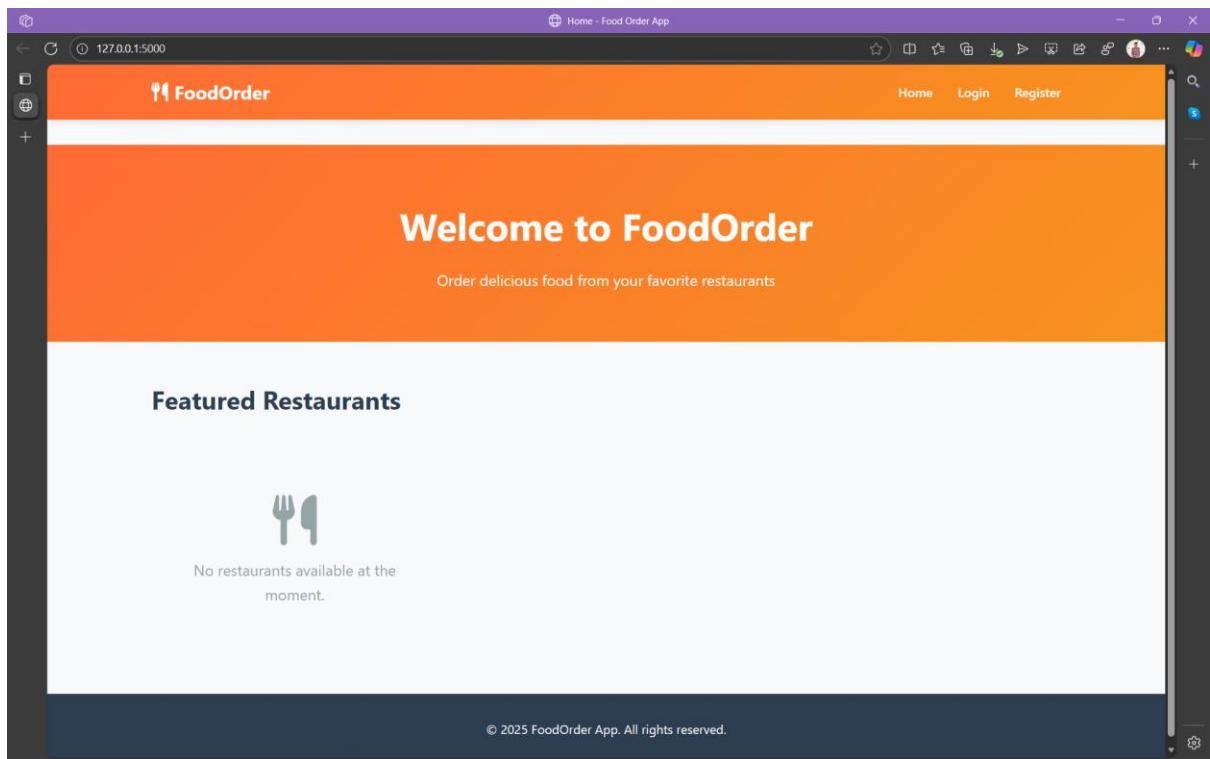
```

Output Screenshots Snip Snap Terminal Bash Command Prompt Shell:-

```

PS C:\Dharshan Raj P A\College\Laboratory\Advanced_Python_Programming_Laboratory\Projects-J_Component\Flask> python .\app.py
Database initialized successfully
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.89.52.1:5000
Press CTRL+C to quit
 * Restarting with stat
Database initialized successfully
 * Debugger is active!
 * Debugger PIN: 289-827-609

```



Login - Food Order App

127.0.0.1:5000/login

FoodOrder

Home Login Register

Registration successful! Please log in. [X](#)

Login

Username: RamKumar

Password:

[Login](#)

Don't have an account? [Register here](#)

© 2025 FoodOrder App. All rights reserved.

Home - Food Order App

127.0.0.1:5000

FoodOrder

Home Login Register

Welcome to FoodOrder

Order delicious food from your favorite restaurants

Featured Restaurants

A large pizza with various toppings like pepperoni, cheese, and vegetables, garnished with fresh herbs and small red berries.

Pizza Paradise

Authentic Italian pizzas with fresh ingredients and traditional recipes. Family-owned since 1995.

📍 123 Main Street, Downtown
📞 +1 (555) 123-4567

[View Menu →](#)

Two large, juicy burgers with lettuce, cheese, and meat patties, served with Coca-Cola glasses in the background.

Burger House

Gourmet burgers made with premium beef and fresh ingredients. Best burgers in town!

📍 456 Oak Avenue, Midtown
📞 +1 (555) 234-5678

[View Menu →](#)

A wooden board with several pieces of nigiri sushi, featuring salmon, avocado, and other toppings, accompanied by wasabi and ginger.

Sushi Master

Fresh sushi and Japanese cuisine. Master chefs prepare authentic dishes daily.

📍 789 Cherry Lane, Uptown
📞 +1 (555) 345-6789

[View Menu →](#)

Home - Food Order App

127.0.0.1:5000

FoodOrder

Home Login Register

owned since 1995.
123 Main Street, Downtown
+1 (555) 123-4567

View Menu →

456 Oak Avenue, Midtown
+1 (555) 234-5678

View Menu →

789 Cherry Lane, Uptown
+1 (555) 345-6789

View Menu →

 Taco Fiesta

Taco Fiesta

Authentic Mexican street tacos and traditional dishes. Spice up your day!

321 Elm Street, Riverside
+1 (555) 456-7890

View Menu →

 Sweet Dreams Bakery

Fresh baked goods, pastries, and desserts. Made fresh daily with love.

654 Maple Drive, Park District
+1 (555) 567-8901

View Menu →

 Noodle Express

Fast and delicious Asian noodles. Perfect for a quick lunch or dinner.

987 Pine Street, Business District
+1 (555) 678-9012

View Menu →

© 2025 FoodOrder App. All rights reserved.

Pizza Paradise - Food Order App

127.0.0.1:5000/restaurant/1

FoodOrder

Home Login Register



Pizza Paradise

Authentic Italian pizzas with fresh ingredients and traditional recipes. Family-owned since 1995.

123 Main Street, Downtown
+1 (555) 123-4567

All Pizza Sides Salads


Margherita Pizza
Classic pizza with fresh mozzarella, tomato sauce, and basil
\$12.99 [Login to Order](#)


Pepperoni Pizza
Traditional pepperoni with mozzarella cheese
\$14.99 [Login to Order](#)


Hawaiian Pizza
Ham, pineapple, and mozzarella cheese
\$15.99 [Login to Order](#)

Pizza Paradise - Food Order App

127.0.0.1:5000/restaurant/1

FoodOrder

Home Cart My Orders Logout (RamKumar)

Pizza Paradise

Authentic Italian pizzas with fresh ingredients and traditional recipes. Family-owned since 1995.
123 Main Street, Downtown
+1 (555) 123-4567

All Pizza Sides Salads



Margherita Pizza
Classic pizza with fresh mozzarella, tomato sauce, and basil
\$12.99 Add to Cart



Pepperoni Pizza
Traditional pepperoni with mozzarella cheese
\$14.99 Add to Cart



Hawaiian Pizza
Ham, pineapple, and mozzarella cheese
\$15.99 Add to Cart

Shopping Cart - Food Order App

127.0.0.1:5000/cart

FoodOrder

Home Cart My Orders Logout (RamKumar)

Shopping Cart

 Margherita Pizza Pizza Paradise \$12.99 each	Quantity: <input type="text" value="1"/> <button>Update</button>	\$12.99 <button>Remove</button>
 Pepperoni Pizza Pizza Paradise \$14.99 each	Quantity: <input type="text" value="1"/> <button>Update</button>	\$14.99 <button>Remove</button>
Order Summary		
Subtotal: \$27.98		
Delivery Fee: \$2.99		
Total:		\$30.97
<button>Proceed to Checkout</button>		

© 2025 FoodOrder App. All rights reserved.

Checkout - Food Order App

127.0.0.1:5000/checkout

FoodOrder

Home Cart My Orders Logout (RamKumar)

Checkout

Delivery Information

Delivery Address
anna Nagar

Phone Number
45656164696

Place Order

Order Summary

Margherita Pizza x 1	\$12.99
Pepperoni Pizza x 1	\$14.99
Subtotal:	\$27.98
Delivery Fee:	\$2.99
Total:	\$30.97

© 2025 FoodOrder App. All rights reserved.

My Orders - Food Order App

127.0.0.1:5000/orders

FoodOrder

Home Cart My Orders Logout (RamKumar)

My Orders

Order #1

November 06, 2025 at 07:50 PM

Total: \$30.97
Items: 2 item(s)
Delivery Address: anna Nagar

Pending

View Details

Order placed successfully! X

© 2025 FoodOrder App. All rights reserved.