Bite & Delight

A Practical Activity Report
Submitted for UI & UX SPECIALIST-(UCS542)
END-Semester Lab Evaluation

Submitted by:
(102303864) Krishiv Goyal

BE Third Year, COE
Group No: 3C62

Submitted to:
Ms. Kanupriya

Computer Science and Engineering Department
TIET, Patiala

INDEX

Introduction	1
Problem Statement	2
Specific Requirements	3-4
Context, Data Flow Diagram	5
System Specification	6-8
Tools Used	9-11
Sample Screenshots	12-14
Output Report	15-16
Conclusion	17
Future Enhancements	18-19

INTRODUCTION

The Bite & Delight Online Restaurant Web Application is a fully responsive, interactive, and user-friendly website designed to modernize the digital food-ordering experience. This project focuses on building a visually appealing platform that allows customers to explore the restaurant's offerings, view reviews, place orders, and interact with an intuitive interface. Developed using HTML, CSS, and JavaScript, the website demonstrates seamless UI/UX design principles, dynamic content handling, and client-side state management using LocalStorage.

The homepage presents an engaging layout featuring high-quality images, smooth navigation, and clear call-to-action elements that guide users toward viewing the menu or placing an order. The services section highlights the restaurant's core strengths, including fast delivery, quality assurance, and easy ordering, while the menu page showcases multiple food items with an "Add to Cart" feature. The design maintains a consistent warm color theme, rounded components, and modern typography to create a welcoming visual identity.

A major feature of this project is its fully functional cart system, which allows users to add multiple items, increase or decrease quantities, remove items, and view real-time billing updates. The cart implements a clean three-column grid layout to ensure proper alignment across various screen sizes. All cart data is stored locally using LocalStorage, enabling a smooth, session-independent shopping experience without requiring a backend.

Another key element is the dynamic review system. Users can submit personalized reviews along with a star rating. Reviews are stored in LocalStorage and immediately displayed on the reviews page inside a horizontally scrollable slider. This system supports multiple review entries, each styled to match the existing testimonial cards. The star-rating interaction is implemented entirely in JavaScript, offering an intuitive click-based selection system.

The website is built with full responsiveness in mind. Through a combination of flexible grids, media queries, and adaptive layouts, the site delivers a consistent and optimized experience across desktops, tablets, and mobile devices. The Google Maps embed ensures that users can easily locate the restaurant, adding another layer of convenience.

Overall, this project integrates aesthetics with functionality, showcasing practical front-end development skills and a strong understanding of user-centric design. It provides an efficient and enjoyable platform that can be extended into a complete online restaurant management system with backend integration in the future.

PROBLEM STATEMENT

Traditional restaurant websites often lack interactivity, mobile responsiveness, and efficient user engagement features such as dynamic reviews or an integrated order management experience. Many customers find it difficult to navigate poorly designed menus, place orders seamlessly, or access essential information like location, pricing, and services. Additionally, small restaurants frequently depend on static pages or third-party delivery platforms, which limit personalization and create dependency on external applications.

There is a need for a modern, responsive, and user-friendly web platform that allows customers to browse the restaurant's menu effortlessly, add or modify items in a cart, and place orders without friction. Furthermore, customers should be able to share their experiences through an easy-to-use review system that updates dynamically without requiring backend support. The platform must maintain visual consistency, load efficiently on all devices, and offer clear navigation to improve overall user satisfaction.

The problem this project addresses is the lack of an integrated, interactive, and efficient online ordering and feedback system for restaurants, specifically one that can operate solely on front-end technologies while still delivering a real-world user experience. By building a fully responsive website with a dynamic cart, star-based review submission, and smooth user interface, this project provides a complete solution that enhances the digital presence and operational convenience of a restaurant.

SPECIFIC REQUIREMENTS

a) Functional Requirements

1.1 Website Navigation

1. The system must allow users to navigate between pages (Home, Menu, Reviews, Find Us, Cart, Review Form).
2. The header navigation bar must remain fixed at the top for quick access.
3. Clicking the cart icon must redirect the user to the order/cart page.

1.2 Menu and Ordering System

1. The system must display all food items with their names, images, and prices.
2. Each menu item must have an Add to Cart button.
3. When a user clicks "Add to Cart", the selected item must be added to the cart with quantity = 1 by default.
4. If the same item is added again, the system must increase its quantity instead of duplicating it.
5. The cart must display:
   - Item name
   - Price per item
   - Quantity
   - Total price per item
   - “Remove” option
6. Users must be able to:
   - Increase item quantity
   - Decrease item quantity
   - Remove items
7. The system must update totals (item count and grand total) in real time.
8. All cart data must be stored in localStorage using the key bite_cart.

1.3 Review System

1. The system must allow users to submit a review through a review form page.
2. The review form must include:
   - Customer name
   - Review message
   - Star rating (1–5)
3. Clicking on stars must activate a visual selection and update the rating.
4. The system must validate that all review fields are filled before submission.
5. Submitted reviews must be stored in localStorage under the key reviews.
6. Submitted reviews must appear dynamically in the Reviews slider on the homepage.
7. The slider must display the:
   - Reviewer name
   - Review text
   - Star rating
8. Dynamic reviews must match the style and size of the static reviews.

1.4 Map & Location

1. The website must display an embedded Google Map showing the restaurant's location.
2. The map must load inside a styled responsive container.

1.5 Persistence

1. Cart items and reviews must persist even after page reload or browser close.
2. Users must be able to clear their cart completely.

b) Non-Functional Requirements

2.1 Performance Requirements

1. The website must load within 2–3 seconds on a standard network.
2. Dynamic UI updates (cart, reviews, slider) must respond within 100 ms.
3. LocalStorage operations must not introduce noticeable delay.

2.2 Usability Requirements

1. The user interface must be intuitive and easy to navigate.
2. The site must follow a consistent color theme (orange + cream tones).
3. Buttons (+/−, Add to Cart, Submit Review) must be clearly visible and accessible.
4. The design must maintain readability using appropriate spacing and typography.

2.3 Responsiveness Requirements

1. The website must be fully responsive on:
    - Mobile phones
    - Tablets
    - Desktop screens
2. Layouts such as menu grid, sliders, header, and cart must adapt automatically to screen size.
3. Interactive elements must remain tappable on mobile (minimum 40px target size).

2.4 Reliability Requirements

1. The system must not lose cart or review data unless the user manually clears it.
2. Errors such as incomplete form submission must be handled gracefully.

2.5 Compatibility Requirements

1. The website must work correctly on:
    - Chrome
    - Firefox
    - Edge
    - Safari
2. The website must function without requiring backend servers.

2.6 Security Requirements

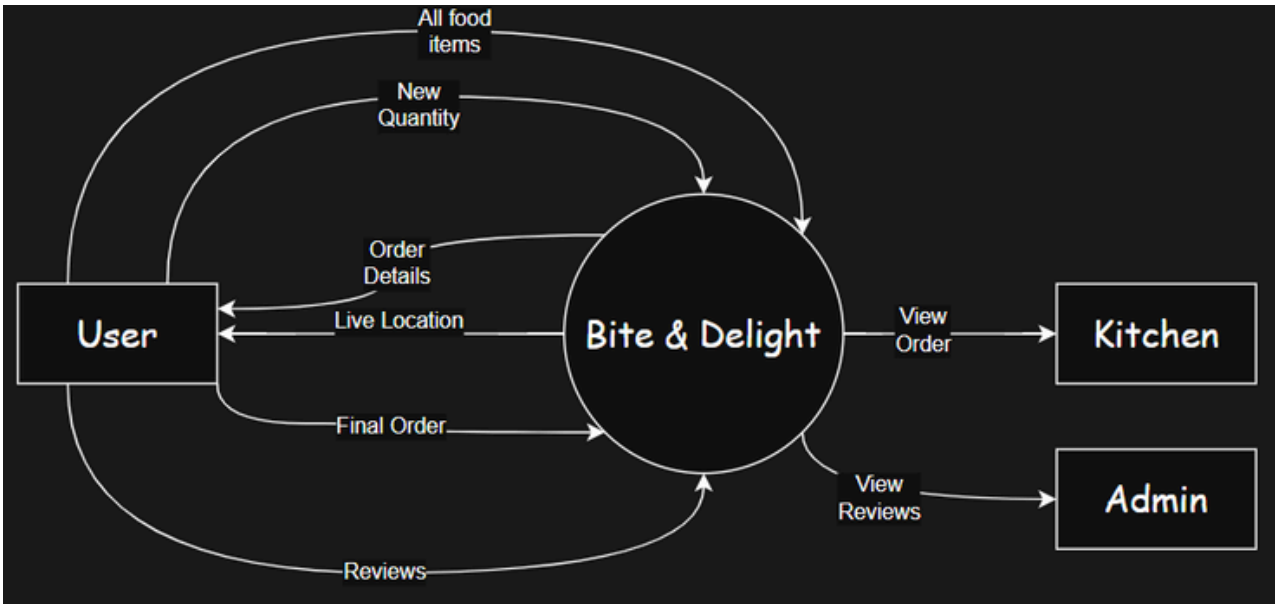
1. User input fields must sanitize text to prevent malformed layout.
2. LocalStorage must not store any sensitive or personal identification data.

2.7 Maintainability Requirements

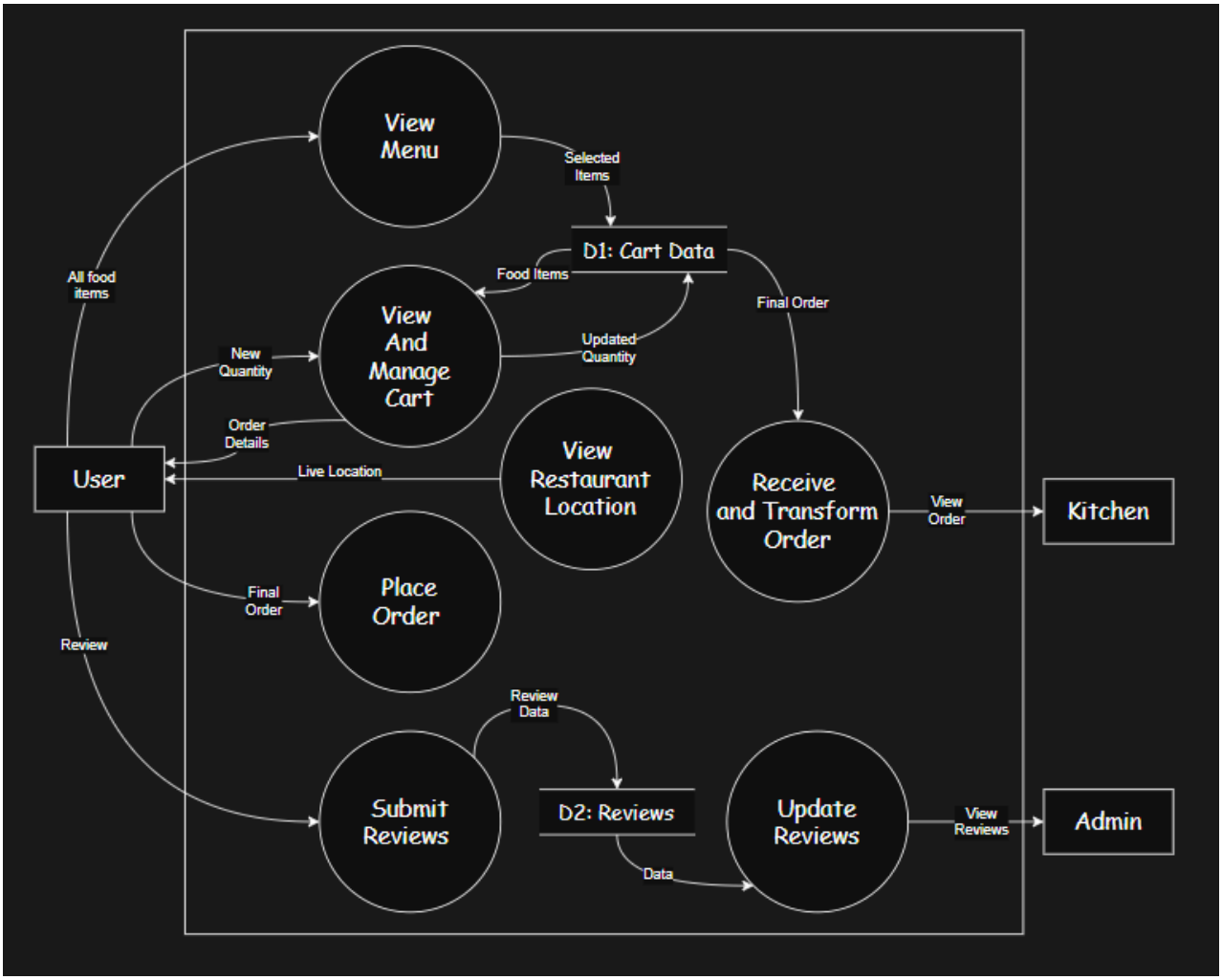
1. Code must be structured clearly to allow easy updates.
2. HTML, CSS, and JavaScript files must be separated for modularity.
3. Naming conventions must be consistent for readability.

CONTEXT LEVEL AND DATA FLOW DIAGRAM

Context Level Diagram -(Level 0)



Data Flow Diagram -(Level 1)



SYSTEM SPECIFICATION

1. Hardware Requirements

Since the project is a lightweight, browser-based web application with no backend server, the hardware requirements are minimal.

1.1 Development System Requirements

- Processor: Dual-core CPU (Intel i3 / AMD equivalent or higher)
- RAM: Minimum 4 GB (8 GB recommended for smoother development)
- Storage: Minimum 500 MB free disk space
- Display: 1366×768 resolution or higher
- Input Devices: Keyboard and Mouse/Touchpad

1.2 End-User (Website Visitor) Requirements

- Device: Any device capable of running a modern browser
    - Smartphones (Android/iOS)
    - Tablets
    - Laptops / Desktops
- RAM: Minimum 1 GB (typical smartphones)
- Processor: Any entry-level processor sufficient
- Internet Connection: Basic broadband or mobile data (3G/4G/5G)

Since the website is optimized for performance, it runs smoothly even on low-end devices.

2. Software Requirements

2.1 Development Software

- Operating System:
  - Windows 10/11
  - macOS
  - Linux (Ubuntu, Fedora, etc.)
- Development Tools:
  - Code Editor: VS Code (recommended), Sublime Text, or similar
  - Browser:
    - Google Chrome (preferred for testing)
    - Firefox
    - Edge
    - Safari
- Design/Testing Utilities:
  - Live Server extension (optional, for auto-refresh)
  - Browser DevTools for debugging

2.2 Runtime Software (For Users)

- Browser with HTML5, CSS3, and JavaScript support, such as:
  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
  - Apple Safari
- No installations required — the website runs directly in the browser.

3. Technology Stack

- Frontend:
  - HTML5 — structure and content
  - CSS3 — layout, responsiveness, and design
  - JavaScript (ES6) — cart logic, dynamic review system, rating functionality
- Client-Side Storage:
  - LocalStorage (for cart and reviews persistence)
- Icons & Fonts:
  - Font Awesome
  - Remix Icons
  - Google Fonts
- Optional Integrations:
  - Embedded Google Maps API (iframe embed)

Tools Used

1. Development Tools

1.1 Visual Studio Code (VS Code)

- Used as the primary code editor.
- Provided features like syntax highlighting, extensions, auto-completion, and Live Server.
- Enhanced productivity through HTML, CSS, and JavaScript debugging support.

1.2 Web Browsers (Chrome, Firefox, Edge)

- Used for testing, debugging, and verifying responsive design.
- Chrome DevTools helped inspect elements, check console logs, analyze layout issues, and debug JavaScript.

2. Front-End Technologies

2.1 HTML5

- Used to build the structural layout of all website pages.
- Implemented sections like Home, Menu, Reviews, Cart, Map, and Review Form.

2.2 CSS3

- Used for styling, page layout, animations, and responsiveness.
- Media queries ensured compatibility across mobile, tablet, and desktop screens.
- Custom styling applied to menu cards, sliders, buttons, cart layout, and theme colors.

2.3 JavaScript (ES6)

- Implemented dynamic behavior and logic.
- Used for:
  - Cart system (add, remove, increase/decrease quantity)
  - Dynamic review submission and display
  - Star rating selection
  - LocalStorage persistence
  - Page routing and interaction handling

3. Supporting Libraries & Frameworks

3.1 Font Awesome

- Used for icons such as:
  - Cart icon
  - Phone icon
  - Social icons (Instagram, WhatsApp, etc.)
  - Star icons in reviews

3.2 Remix Icon

- Additional icon pack used to enhance UI elements.

3.3 Google Fonts (Poppins & EB Garamond)

- Used for typography across the website.
- Improved readability and modern appearance.

4. Additional Tools

4.1 Google Maps Embed

- Integrated via iframe to display restaurant location.

4.2 LocalStorage (Browser API)

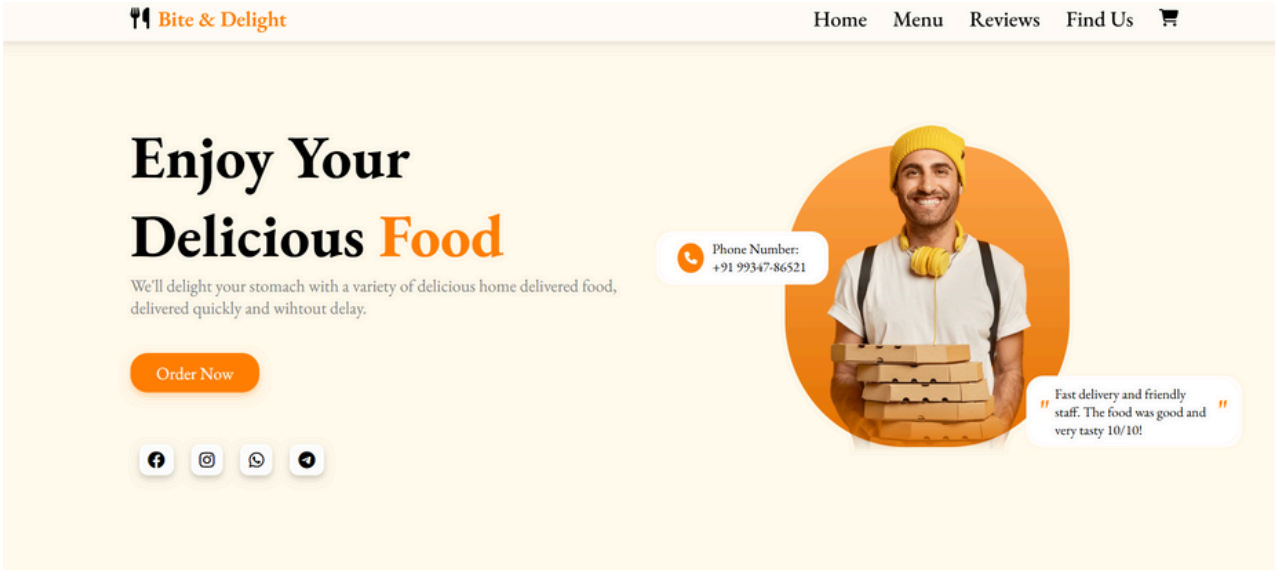
- Used to store:
  - Cart items
  - User-submitted reviews
- Provided persistent data without requiring a backend or database.

4.3 Git / GitHub (Optional if used)

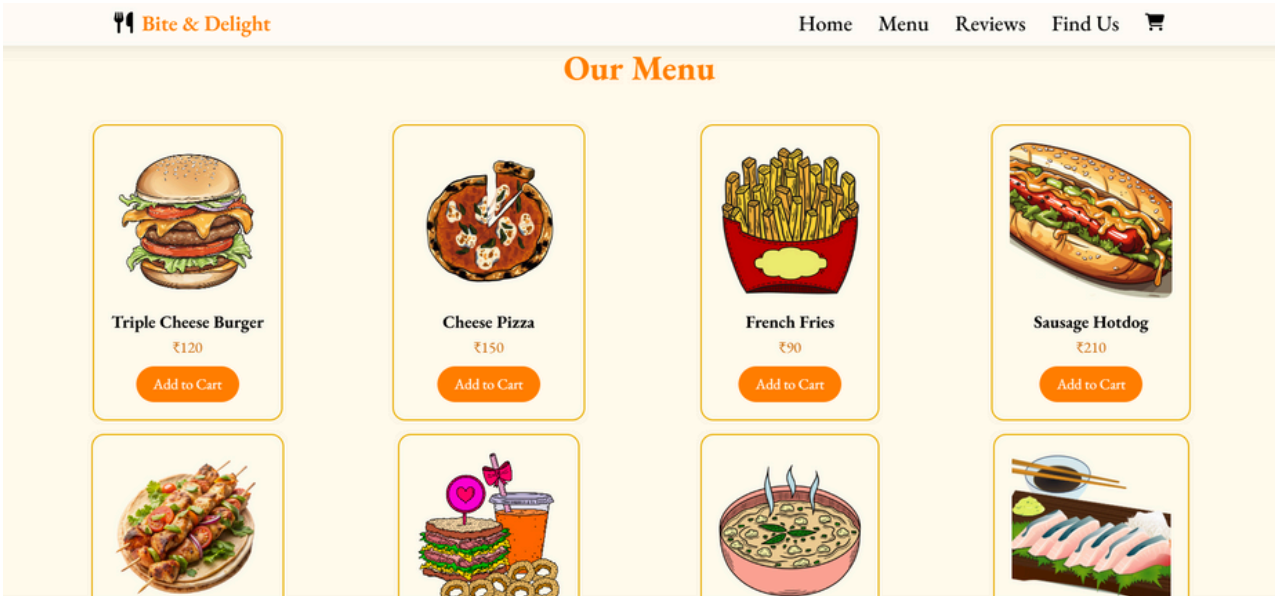
- Could be used for version control and backup.
- Helps track changes and maintain project history.

SAMPLE SCREENSHOTS

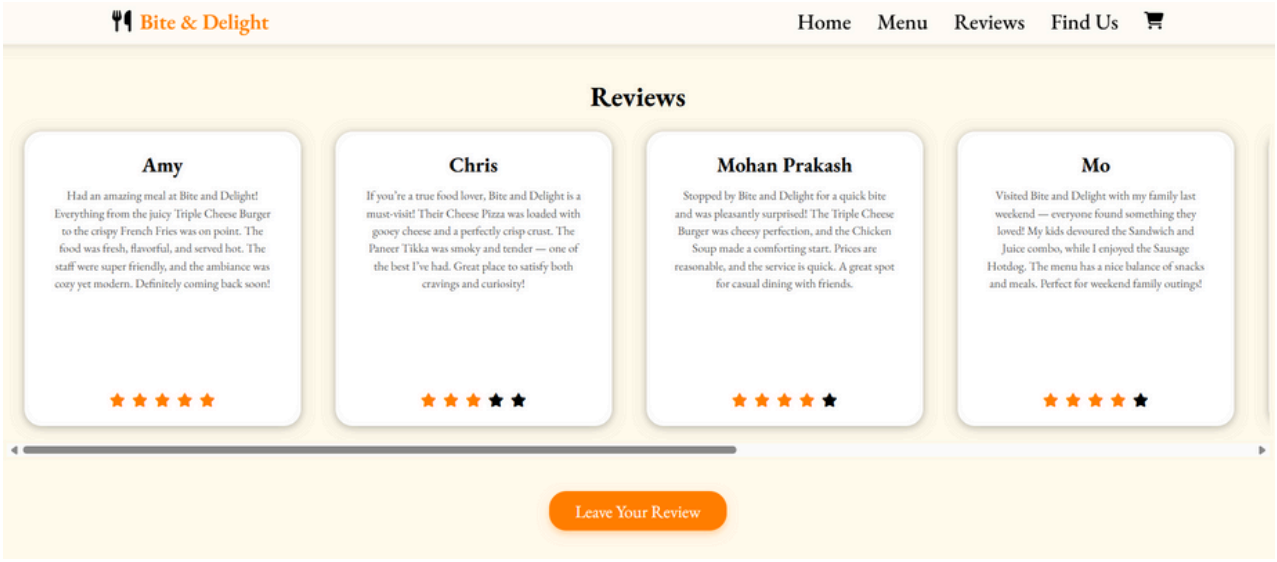
1) Home Section



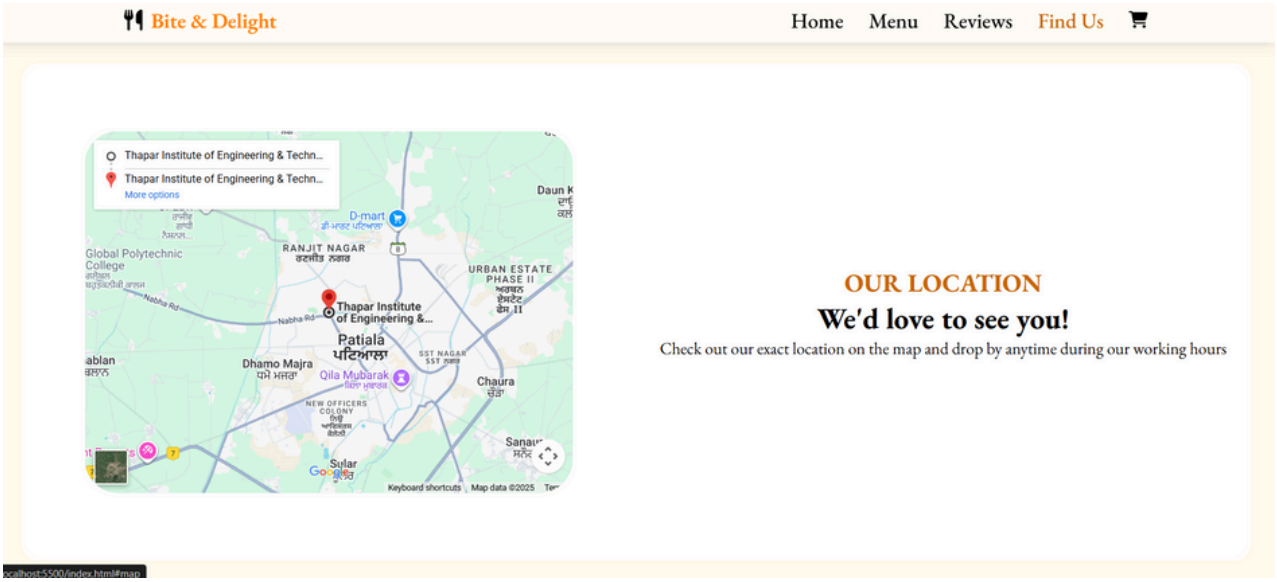
2) Menu Section



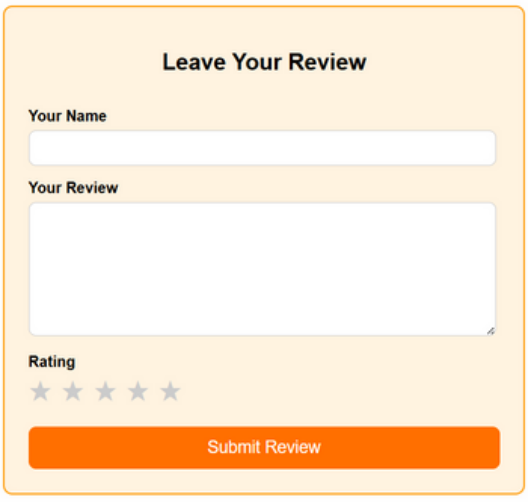
3) Review Section



4) Find Us Section



5) Leave Review Page



6) Cart Page


Bite & Delight
Home
Menu
Reviews
Find Us
6
Your Order
Triple Cheese Burger
₹120 each
-
1
+
₹120
Remove
Chicken Soup
₹70 each
-
2
+
₹140
Remove
Sausage Hotdog
₹210 each
-
1
+
₹210
Remove
Sandwich And Juice
₹220 each
-
2
+
₹440
Remove
Total Items: 6
Total: ₹910
Clear Cart
Place Order

OUTPUT REPORTS

This section outlines the key reports generated during the development and testing phases of the Bite & Delight Online Restaurant Website. While the actual documents are not included, the details below summarize the processes followed and the insights gained.

a. Functionality Testing Report

Since the project is fully front-end based (HTML, CSS, JavaScript, LocalStorage), functionality testing was carried out manually across multiple browsers and devices. The key interactive components tested include:

1. Add to Cart Functionality
   - Verified that each menu item correctly adds to the cart.
   - Confirmed that repeated additions increase the quantity instead of creating duplicate entries.
   - Ensured that item totals and overall cart totals update dynamically.
   - LocalStorage (bite_cart) was checked to confirm that data persists across sessions.
2. Quantity Update (Increment/Decrement)
   - Confirmed that "+" increases quantity and "−" decreases quantity.
   - Validated that decreasing quantity at 1 removes the item from the cart.
   - Ensured real-time update of the total amount.
3. Remove Item Function
   - Verified that the "Remove" button deletes the correct product.
   - Confirmed that LocalStorage updates immediately after removal.
4. Review Submission System
   - Tested the star rating selection to ensure accurate value capture.
   - Confirmed that review form validation prevents empty submissions.
   - Verified that submitted reviews save correctly in LocalStorage (reviews).
   - Ensured new reviews appear in the homepage slider dynamically, matching the same design as static reviews.
5. Responsive Design Testing

Tested across:

- Mobile devices (Android/iOS)
- Tablets
- Desktop screens

Key checks:

- Navbar alignment
- Menu grid adjustments
- Slider scroll behaviour
- Cart layout grid consistency
- Button touch-responsiveness
- Image scaling and text wrapping

All tests confirmed smooth adaptability across screen sizes.

b. Debugging Logs

During development, debugging was performed extensively to ensure error-free functionality and consistent UI behaviour. The following methods were used:

1. Console Logs

Console logging was used to track:

- Star rating values during review submission
- Cart updates (add, remove, increment, decrement)
- LocalStorage read/write operations
- Slider dynamic rendering behaviour
- Responsive mode detection

These logs helped trace issues in real-time.

2. Error Handling & Validations

Multiple safeguards were implemented:

- Form validation for review submission
- Empty-cart check before rendering
- Validation for missing fields
- Protection against invalid star ratings
- Handling of malformed LocalStorage data

3. UI Debugging

Developer tools were used to inspect:

- Flexbox and grid alignment
- Overflow issues in the review slider
- Spacing and sizing inconsistencies
- Mobile view text overflow
- Z-index mismatches in header and overlay elements

4. Layout & Responsiveness Debugging

Using Chrome DevTools and Firefox Responsive View:

- Ensured menu cards scale properly
- Fixed misaligned slider items
- Corrected cart grid to maintain 3-column structure

These debugging steps ensured a visually stable and highly responsive website.

Conclusion

The Bite & Delight Online Restaurant Website successfully demonstrates how modern front-end technologies can be combined to create a responsive, dynamic, and user-friendly digital platform without relying on a backend server. The project fulfills its primary objective of providing customers with an intuitive way to browse the menu, add items to a smart cart system, submit reviews, and access essential restaurant information through an attractive and interactive interface.

Throughout the development process, the project focused on usability, efficiency, and visual consistency. The responsive layout ensures that users on mobile phones, tablets, or desktops receive an optimized viewing experience, while features like smooth navigation, structured menu presentation, and visually appealing components contribute to enhanced user satisfaction. The cart system, backed by LocalStorage, enables real-time quantity updates, item removal, and persistent order state even after page reloads, demonstrating effective client-side data handling.

The dynamic review system further strengthens the interactive aspect of the website. By allowing users to submit personalized feedback accompanied by star ratings, the platform encourages engagement and builds credibility. These reviews appear instantly within the homepage testimonial slider, showing seamless DOM manipulation and effective use of LocalStorage for storing and retrieving user-generated content. The Google Maps integration provides a convenient way for customers to locate the restaurant, complementing the overall experience.

Comprehensive testing, debugging, and refinement helped improve both the functional and aesthetic aspects of the platform. Browser testing validated consistent performance, while UI debugging ensured proper alignment, spacing, and responsiveness across all devices. The final system is visually appealing, technically sound, and feature-rich, offering a complete front-end solution suitable for real-world restaurant applications.

In conclusion, this project not only showcases proficiency in HTML, CSS, and JavaScript but also highlights the importance of user-centered design, responsive layouts, and interactive elements. The Bite & Delight website serves as a strong foundation that can be expanded in the future with backend integration, user authentication, online payments, and order tracking—ultimately evolving into a fully functional restaurant management and delivery system.

Future Enhancements

Although the current version of the Bite & Delight Online Restaurant Website successfully delivers a responsive and interactive front-end experience, several enhancements can be implemented to transform it into a fully functional, production-level system. The following future improvements will significantly expand its capabilities, security, and usability.

1. Backend Integration with Database

One of the most important upgrades is integrating a backend system using technologies such as Node.js and Express. A relational or NoSQL database (MySQL, MongoDB, etc.) can be used to store:

- User accounts and login details
- Menu items and pricing
- Orders and transaction records
- Customer reviews and ratings
- Discount and loyalty program data

Database integration will also ensure that data is securely handled, persistent across devices, and can be updated in real time.

2. User Authentication (Login/Signup System)

Implementing a secure login and registration system will allow users to create personal accounts. Features include:

- Secure password hashing
- Email or phone verification
- User roles (customer/admin)
- Password reset functionality

Registered customers can track their past orders, save delivery addresses, and manage personal profiles. Authentication will also prevent unauthorized access to sensitive or restricted pages.

3. Online Order Placement System

The cart system can be extended to a fully functional order placement module. Future upgrades include:

- Order confirmation page
- Delivery address collection
- Payment gateway integration (UPI, cards, wallets)
- Order ID generation
- Real-time order tracking
- Notifying kitchen/admin via dashboard

This enhancement will convert the project into a complete online food ordering platform

4. Special Discount & Loyalty System

To improve customer engagement, a dynamic discount system can be introduced:

- Loyalty points based on total purchase amount
- Special discount codes during festivals or events
- Referral bonuses
- Membership tiers (Silver, Gold, Platinum)
- Personalized offers based on order history

This feature will encourage returning customers and increase sales.

5. Admin Dashboard

An advanced admin panel can be created with the following capabilities:

- Add, update, or remove menu items
- Monitor orders in real time
- Manage reviews (approve/remove)
- Track customer activity
- Generate sales reports
- Apply discounts or offers

This backend tool will help restaurant staff manage the platform efficiently.

6. Improved Review Management

The existing LocalStorage system can be upgraded to a database-backed review system featuring:

- Review moderation
- Ability to reply to user feedback
- Rating analytics (average ratings, most-liked items)
- Verified-customer reviews

This will improve credibility and data accuracy.

7. Enhanced UI/UX Features

Additional interface improvements may include:

- Dark mode / theme switcher
- Improved animations and transitions

These enhancements will enrich the overall user experience.

8. Deployment & Hosting

Finally, future versions can be hosted on platforms such as:

- AWS
- Vercel
- Netlify
- Render

For automatic updates and stable performance.