Front End Engineering - II

Project Report

Semester-IV (Batch-2022)

Tomato (A food delivery website)

Supervised By:

Dr. Baljit Kaur

Submitted By:

Karan Kumar Garg: 2210990478 (G-10)

Joyash Sood: 2210990466 (G-10)

Jatin Saini: 2210990460 (G-10)

Kartavya Tomar: 2210990461 (G-10)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Table of Content

1. Introduction

In the hustle and bustle of modern life, managing our daily tasks efficiently has become paramount. Among these tasks, the management and delivery of food stand out as areas where innovation can significantly improve our daily lives. Enter **Tomato**: a groundbreaking web-based platform designed to revolutionize how we interact with food.

Tomato is more than just a food delivery service; it is a comprehensive solution that seamlessly integrates food management with delivery, all within a user-friendly digital interface. By harnessing the power of cutting-edge technologies like React, Node.js, and MongoDB, Tomato offers a unique blend of convenience, efficiency, and sustainability.

This introduction will delve into the background of Tomato, outlining its objectives and highlighting its significance in the realm of food management and delivery. From reducing food waste to promoting healthier eating habits, Tomato aims to redefine the way we approach food in our daily lives. So let's explore how Tomato is poised to transform the food landscape for the better.

1.1 Background

In today's fast-paced world, convenience and efficiency in managing daily tasks have become a priority for many individuals. One such area where this demand is particularly high is food management and delivery. With the rise of urban living, a significant portion of the population seeks solutions that provide quick access to meals, easy management of food resources, and seamless delivery services. Enter **Tomato**, a comprehensive web-based platform designed to revolutionize the way we manage and receive food. Built using the powerful and flexible technologies of React, Node.js, and MongoDB, Tomato aims to provide users with an intuitive and efficient experience in food management and delivery.

1.2 Objectives

The primary objectives of Tomato are:

1. **Efficient Food Management**: To offer users an intuitive interface to manage their food inventory, track expiration dates, and reduce food waste.
2. **Seamless Delivery Service**: To facilitate a fast and reliable food delivery system that connects users with a wide range of local restaurants and food suppliers.
3. **User-Friendly Experience**: To ensure that the platform is easy to navigate, providing a seamless experience from browsing menus to placing orders.
4. **Integration of Technology**: To leverage the latest in web technologies to provide a robust, secure, and scalable service.
5. **Sustainability**: To promote sustainable practices by encouraging efficient food use and reducing waste.

1.3 Significance

The significance of Tomato lies in its potential to address several key challenges faced by modern urban dwellers. By combining food management and delivery into a single platform, Tomato not only enhances user convenience but also promotes sustainability.

1. **Reduction of Food Waste**: Through effective food management features, users can keep track of their inventory and expiration dates, leading to a significant reduction in food waste.
2. **Time-Saving**: The platform streamlines the process of meal planning and ordering, saving users valuable time that can be spent on other activities.
3. **Improved Access**: Tomato connects users with a variety of local food providers, enhancing their access to diverse culinary options.
4. **Technological Advancement**: By utilizing React, Node.js, and MongoDB, Tomato represents the cutting edge of web development, providing a reliable and scalable solution.

5. **Health and Wellbeing**: With easy access to a variety of food options, users can make healthier food choices and manage their diet more effectively.

In summary, Tomato is not just another food delivery service; it is an innovative platform designed to simplify food management and delivery, making it an indispensable tool for the modern consumer.

2. Problem Definition and Requirements

In today's fast-paced world, individuals often struggle with several challenges related to food management and delivery:

1. **Disorganized Food Management**: Many people find it difficult to keep track of their food inventory, leading to food spoilage and waste.
2. **Limited Access to Variety**: Traditional food delivery services may offer limited options, restricting users' choices and culinary experiences.
3. **Inefficient Delivery Services**: Some delivery services may lack reliability or speed, leading to frustration among users.
4. **Lack of Integration**: Existing platforms may focus solely on food delivery or management, lacking seamless integration between the two aspects.

2.1 Software Requirements

Development Environment

Languages: HTML, CSS, JavaScript, NodeJS, React

IDEs: VS Code

2.2 Hardware Requirements

Development Environment

Processor: Multi-core processor (e.g., Intel Core i4 or higher)

RAM: At least 4GB for handling datasets and models

Storage: SSD storage with sufficient space for datasets and development environments

3. Proposed Design / Methodology

The design of YummyBites revolves around creating a seamless and user-friendly experience that integrates food management and delivery services. This involves careful consideration of both the frontend and backend components, as well as the overall architecture of the platform.

3.1 Frontend Design

3.1.1 **React Framework**: Tomato utilizes the React framework for its frontend development, enabling the creation of dynamic and interactive user interfaces. React's component-based architecture allows for the modular design of different features, facilitating easier maintenance and scalability.

3.1.2 **Responsive Design**: The frontend of Tomato is designed to be responsive, ensuring optimal performance and usability across various devices and screen sizes. This includes desktop computers, laptops, tablets, and smartphones, providing users with a consistent experience regardless of the device they use.

3.1.3 **Intuitive User Interface**: The user interface of Tomato is designed to be intuitive and easy to navigate. Clear navigation menus, prominent call-to-action buttons, and intuitive search functionalities help users find what they need quickly and efficiently.

3.2 Backend Design

3.2.1 **Node.js and Express.js**: Tomato leverages the Node.js runtime environment along with the Express.js framework for building its backend infrastructure. Node.js enables non-blocking, event-driven architecture, making it well-suited for handling concurrent requests and real-time interactions.

3.2.2 **MongoDB Database**: MongoDB serves as the backend database for Tomato, offering a flexible and scalable solution for storing and managing data. Its document-oriented nature allows for the storage of complex data structures, making it ideal for handling diverse information such as user profiles, food inventory, and order history.

3.2.3 **RESTful API**: Tomato follows the principles of Representational State Transfer (REST) architecture for its API design. This enables seamless communication between the frontend and backend components, allowing for the exchange of data in a standardized and efficient manner.

3.3 Methodology

3.3.1 **Agile Development**: Tomato adopts an agile development methodology, emphasizing iterative development and continuous feedback. This allows for the rapid delivery of features and enhancements, ensuring that the platform remains responsive to changing user needs and market trends.

3.3.2 **User-Centered Design**: The development process of Tomato is guided by a user-centered design approach, focusing on understanding the needs and preferences of its target audience. User feedback is regularly solicited and incorporated into the design and development process, ensuring that the platform meets the expectations of its users.

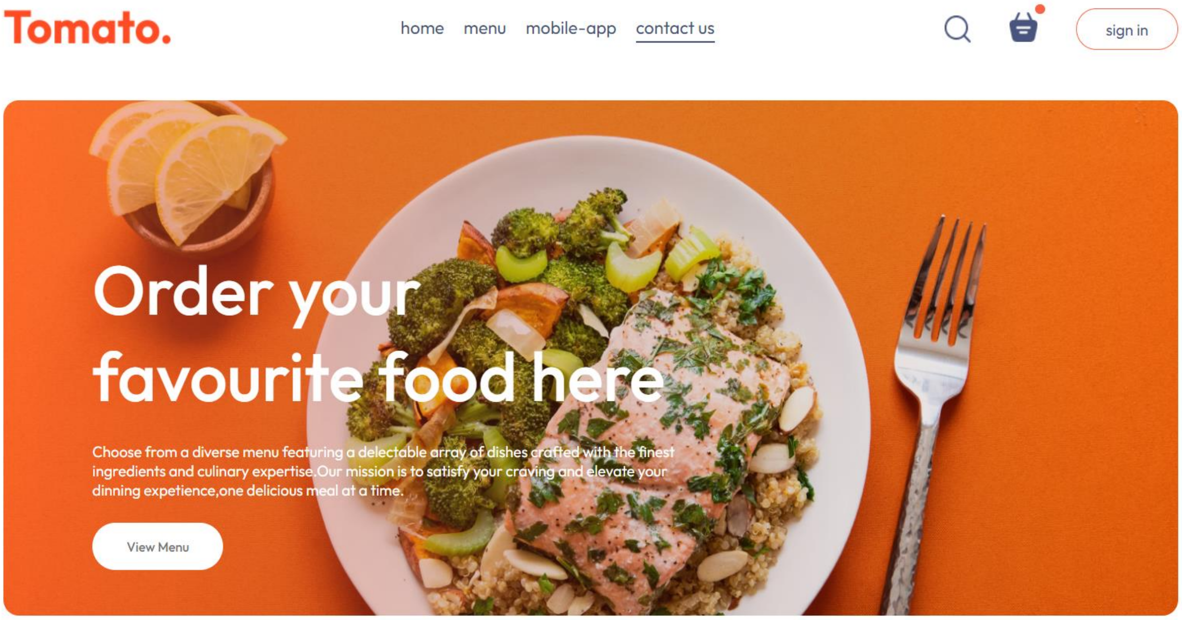
3.3.3 **Continuous Integration and Deployment (CI/CD)**: Tomato employs CI/CD practices to automate the testing, integration, and deployment of code changes. This helps maintain code quality, reduces the risk of errors, and enables rapid deployment of new features and updates.

3.3.4 **Scalability and Performance Optimization**: Tomato is designed with scalability and performance in mind, employing techniques such as caching, load balancing, and horizontal scaling to ensure that the platform can handle increased traffic and user demand without compromising performance.

By adopting these design principles and methodologies, Tomato aims to create a robust and reliable platform that delivers a seamless and satisfying experience for its users, from food management to delivery.

4. Results

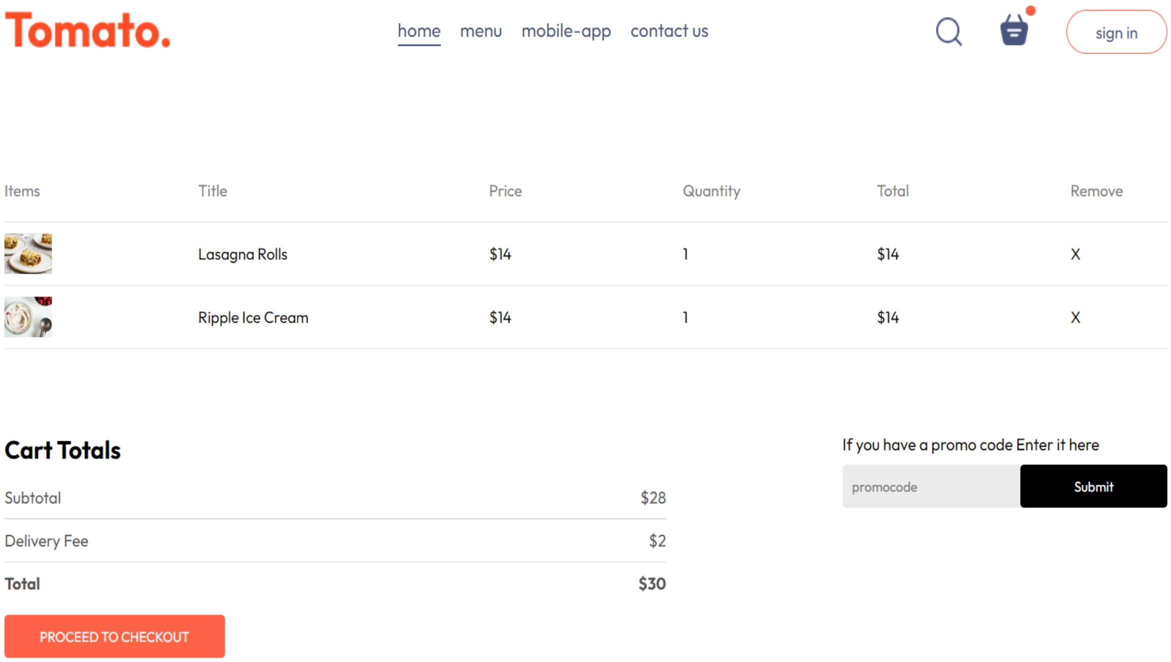
4.1 This is the front page



Order your favourite food here

Choose from a diverse menu featuring a delectable array of dishes crafted with the finest ingredients and culinary expertise.Our mission is to satisfy your craving and elevate your dinning expetience,one delicious meal at a time.

4.2 This is the payment page/ checkout page



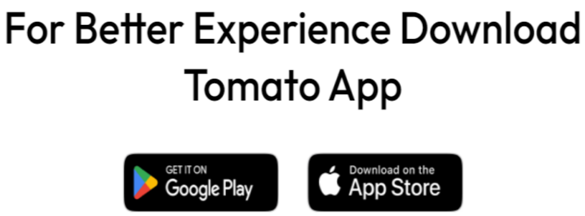
Items	Title	Price	Quantity	Total	Remove
	Lasagna Rolls	$14	1	$14	X
	Ripple Ice Cream	$14	1	$14	X

Cart Totals

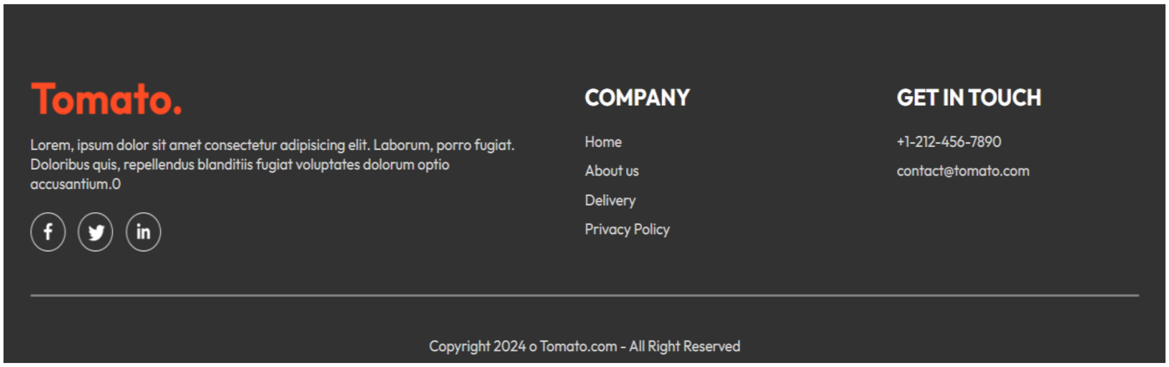
Subtotal	$28
Delivery Fee	$2
Total	**$30**

If you have a promo code Enter it here

4.3 This is the Contact Us



For Better Experience Download Tomato App



Tomato.

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Laborum, porro fugiat. Doloribus quis, repellendus blanditiis fugiat voluptates dolorum optio accusantium.0

COMPANY

- Home
- About us
- Delivery
- Privacy Policy

GET IN TOUCH

+1-212-456-7890

contact@tomato.com

Copyright 2024 o Tomato.com - All Right Reserved

5. References

1. geeksforgeeks.com
2. W3schools.com
3. Codepen.com
4. React.org
5. Devdocs.io
6. Freecodecamp.org
7. MDN web-docs