Tomato.

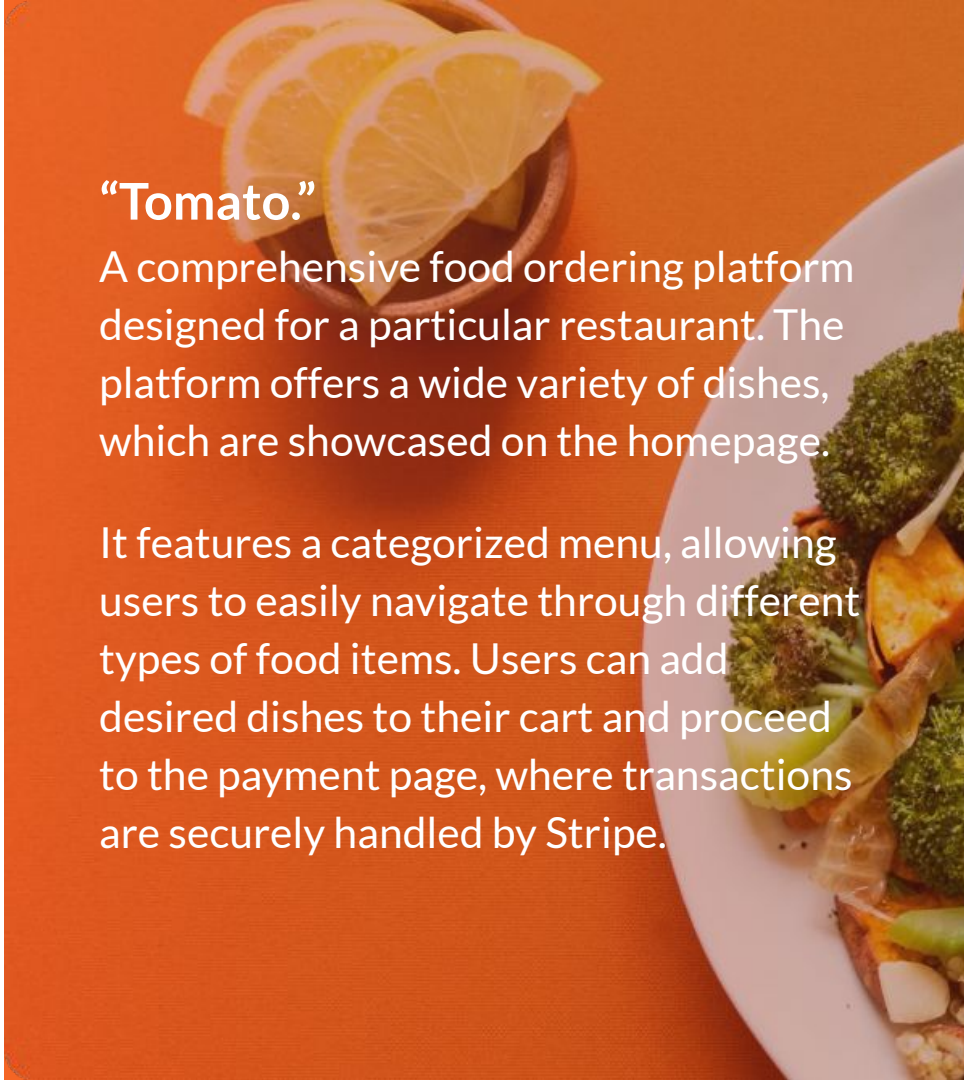
Project Report: Summer Internship

M.Sc. (Information Technology) 2nd Semester

Name - Mayank Tailor
S_ID - 202312052
Guided By - Dr. Shruti Bhilare

Tomato.

Overview



“Tomato.”

A comprehensive food ordering platform designed for a particular restaurant. The platform offers a wide variety of dishes, which are showcased on the homepage.

It features a categorized menu, allowing users to easily navigate through different types of food items. Users can add desired dishes to their cart and proceed to the payment page, where transactions are securely handled by Stripe.

Functional Requirements

User Functions

- Users should be able to log in and log out of their accounts.
- Users should be able to browse food options and search for specific items.
- Users should be able to add items to their shopping cart and specify quantities.
- Users should be able to check out, enter delivery details, and place orders.
- Users should be able to track the status of their orders.
- Users should be able to check his previous orders.
- Non-logged-in users should be prompted to login to access checkout.

Functional Requirements

Admin Functions

- Admins should be able to view incoming orders and manage them.
- Admins should be able to accept orders and mark them as out for delivery.
- Admins should be able to manage food items in the inventory, including adding, deleting, and updating items.
- Admins should be able to view past orders and earnings.
- Admins should be able to accept or reject items for inclusion in the inventory.
- Admins should be able to indicate when an item is ready for delivery.

Non-Functional Requirements

Security: The system should ensure secure transactions and protect user data.

Usability: The system should be user-friendly and easy to navigate.

Scalability: The system should be scalable to accommodate future growth.

Maintainability: The system should be designed in a way that makes it easy to maintain and update.

Responsive: The system should be responsive and should work on every screen size.

Coding & Design

Tomato.

Technologies and Frameworks

Front-End

- **Vite React Framework:** Used to create a fast and modular user interface.
- **JavaScript:** The primary programming language for front-end development.
- **CSS3 and HTML5:** Used for styling and structuring web pages.
- **Axios:** A promise-based HTTP client for making API requests.
- **Toastify:** For displaying notifications and alerts.

Tomato.

Back-End

- **Node.js:** A JavaScript runtime for building the server-side of the application.
- **Express.js:** A web application framework for Node.js, used for client/server connectivity and handling routing and middleware.
- **MongoDB Atlas:** A cloud-based NoSQL database for storing application data.
- **Mongoose:** An ODM (Object Data Modeling) library for MongoDB, used to define schemas and interact with the database.
- **JsonWebToken:** Used to create and verify web tokens for authentication.
- **Bcrypt:** A library for hashing passwords to ensure secure user authentication.
- **Stripe:** A payment gateway for processing payments securely.
- **Validator:** Used to validate email addresses and other input data.
- **Nodemon:** Automatically restarts the server when changes are made to the codebase.
- **Envdev:** For managing environment variables and configuration.

Tomato.

Testing & API Integration

APIs : Multiple APIs were developed to handle various functionalities within the app. For example some key APIs are:

- POST /api/user/logout: Handles user logout.
- POST /api/user/register: Registers new users.
- GET /api/food/list: Fetches all available food items.
- POST /api/food/add: Adds a new food item (admin only).

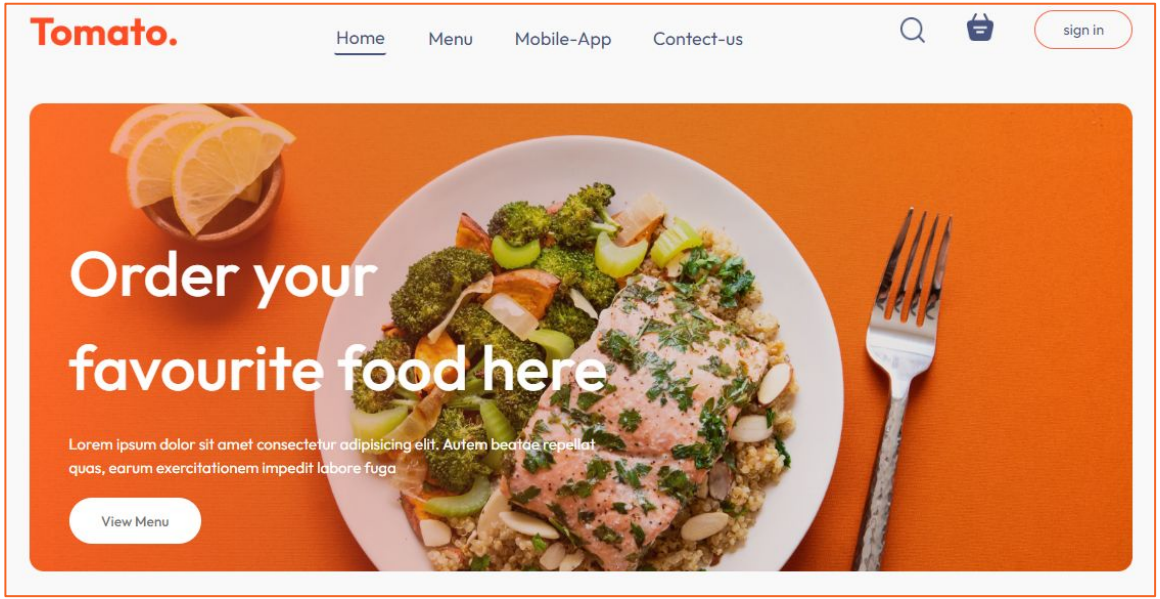
Testing : To ensure the reliability and performance of the "Tomato" web application, various testing methods were employed

- Unit Testing :
- Integration Testing
- End-to-End (E2E) Testing
- Performance Testing

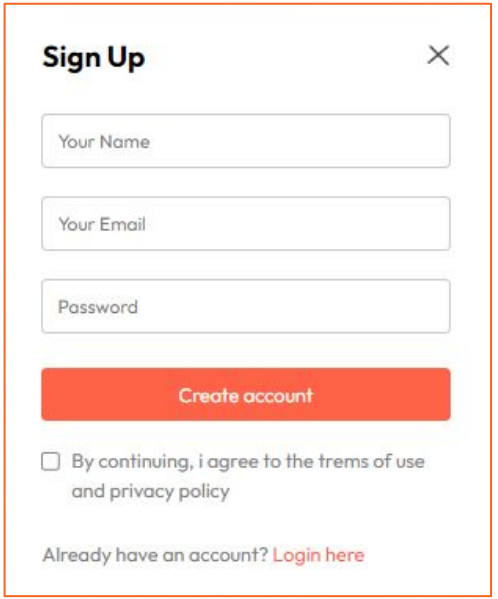
ScreenSorts

Tomato.

Home Page:



Sign-Up Page:



Tomato.

Explore Our Menu

Lorem ipsum dolor sit amet consectetur adipisicing elit. Animi quaerat laboriosam eum quod dolor! Doloribus.


Salad


Rolls


Deserts


Sandwich


Cake

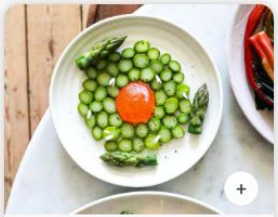

Pure Veg


Pasta

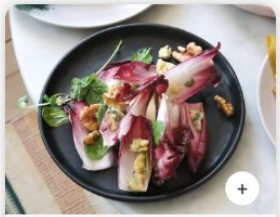

Noodles

Menu

Top Dishes Near You


Greek salad ★★★★☆
₹175


Veg salad ★★★★☆
₹145

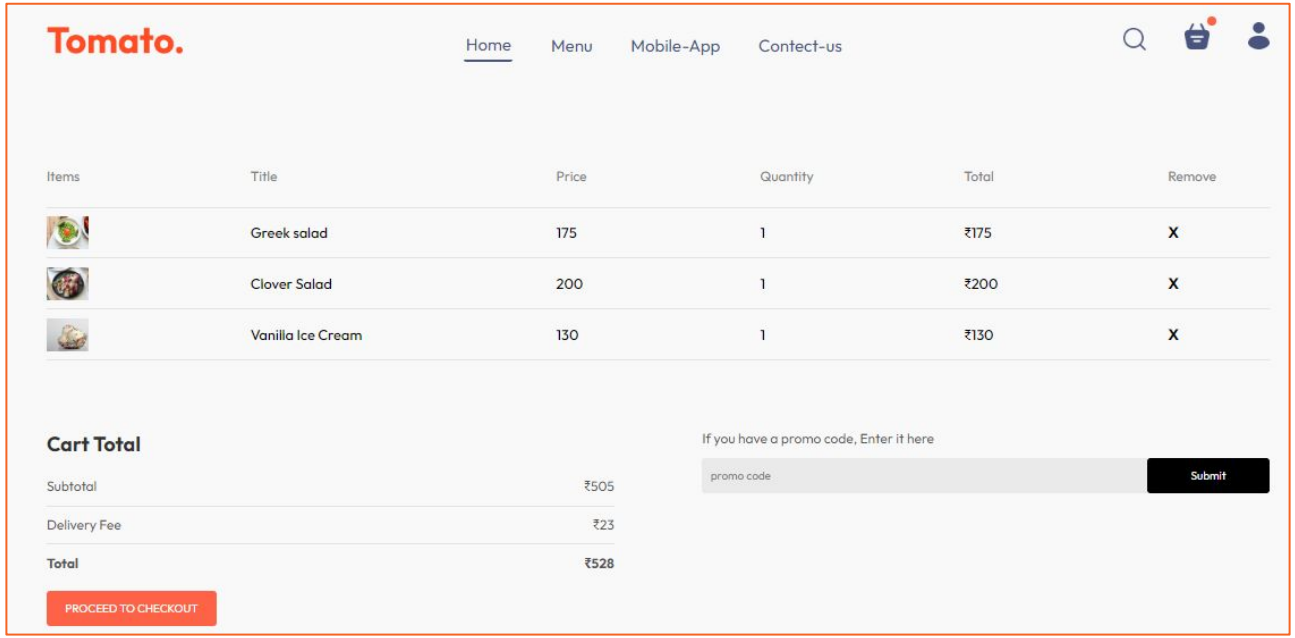

Clover Salad ★★★★☆
₹200


Chicken Salad ★★★★☆
₹210

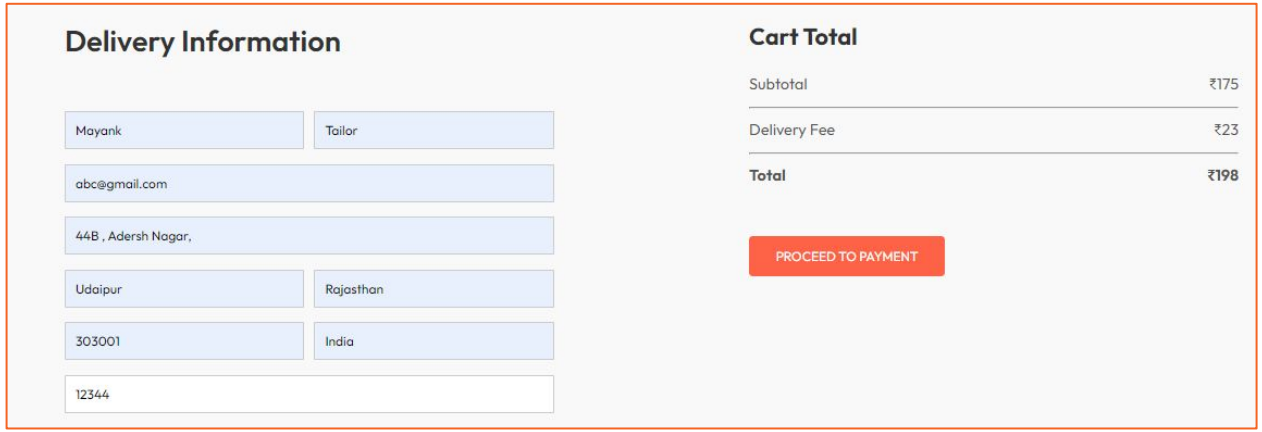
Food Cards

Tomato.

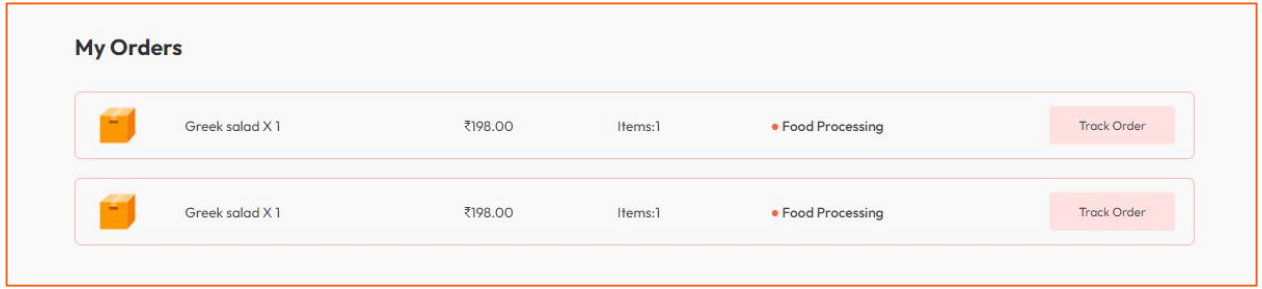
Cart


Tomato.
Home
Menu
Mobile-App
Contect-us
Items
Title
Price
Quantity
Total
Remove
Greek salad
175
1
₹175
X
Clover Salad
200
1
₹200
X
Vanilla Ice Cream
130
1
₹130
X
Cart Total
Subtotal
₹505
Delivery Fee
₹23
Total
₹528
PROCEED TO CHECKOUT
If you have a promo code, Enter it here
promo code
Submit

Tomato.



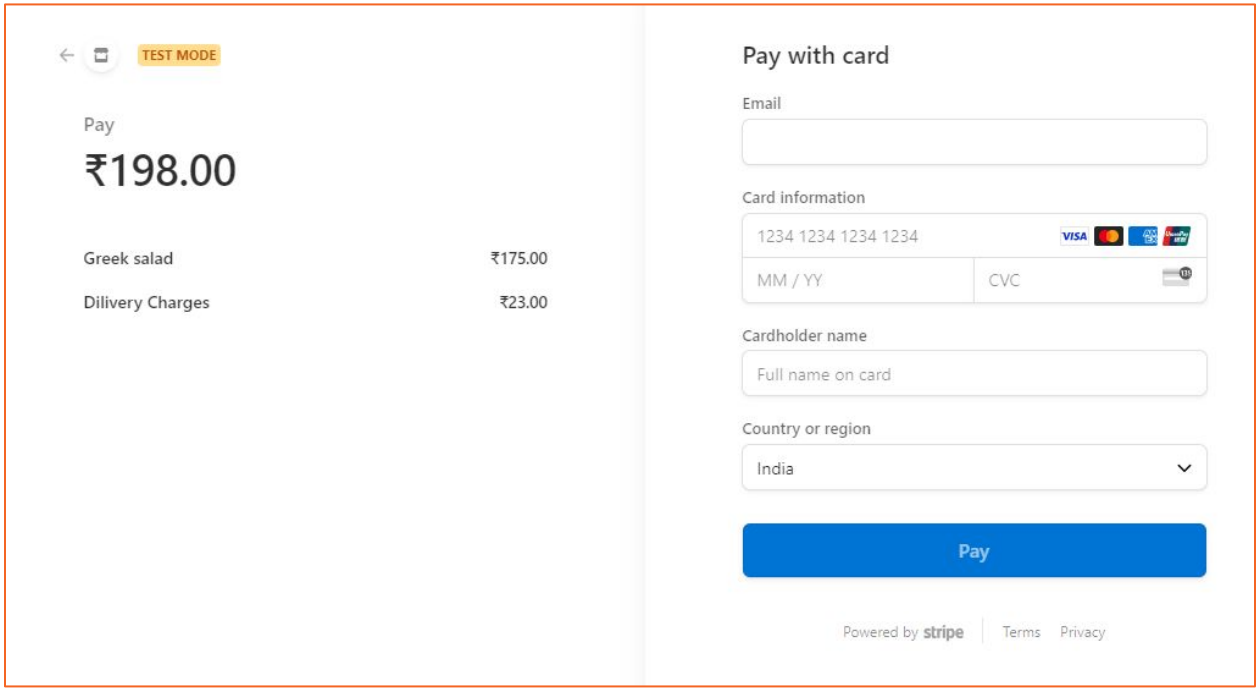
Order's Page



My Orders Page

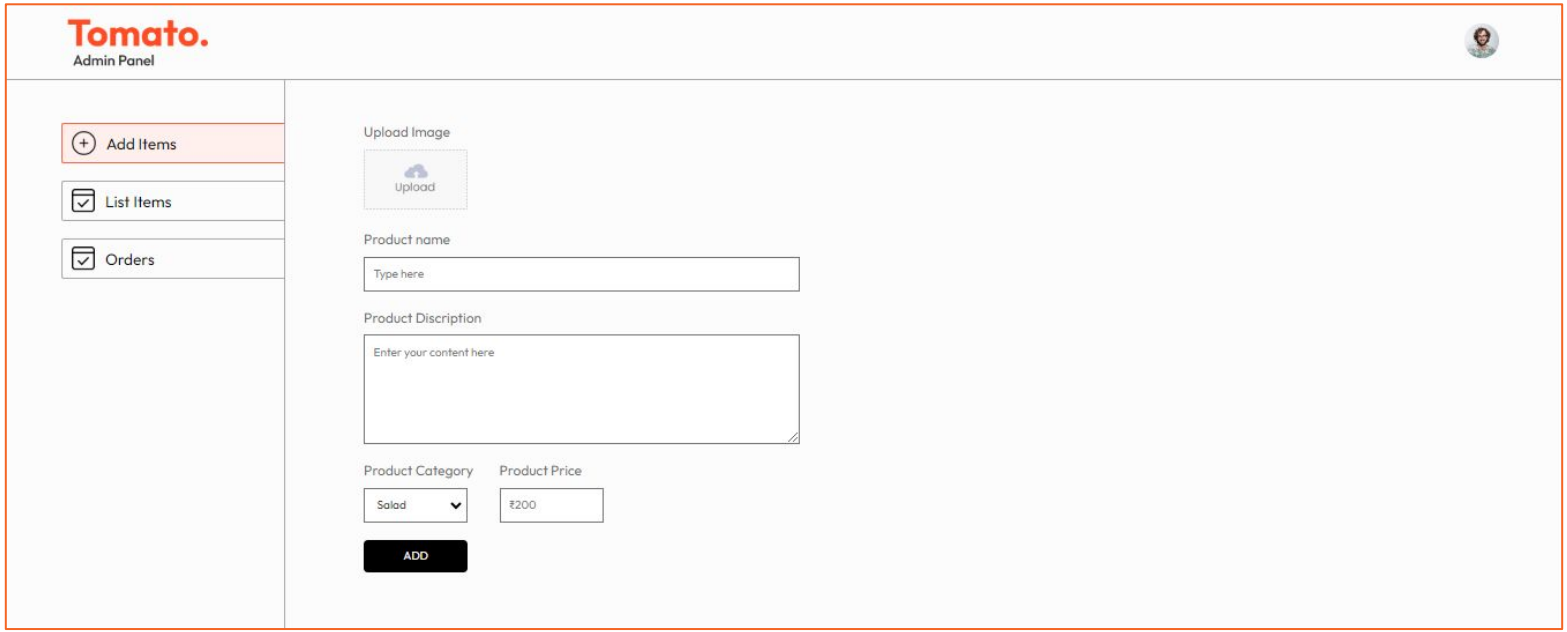
Tomato.

Payment Gateway



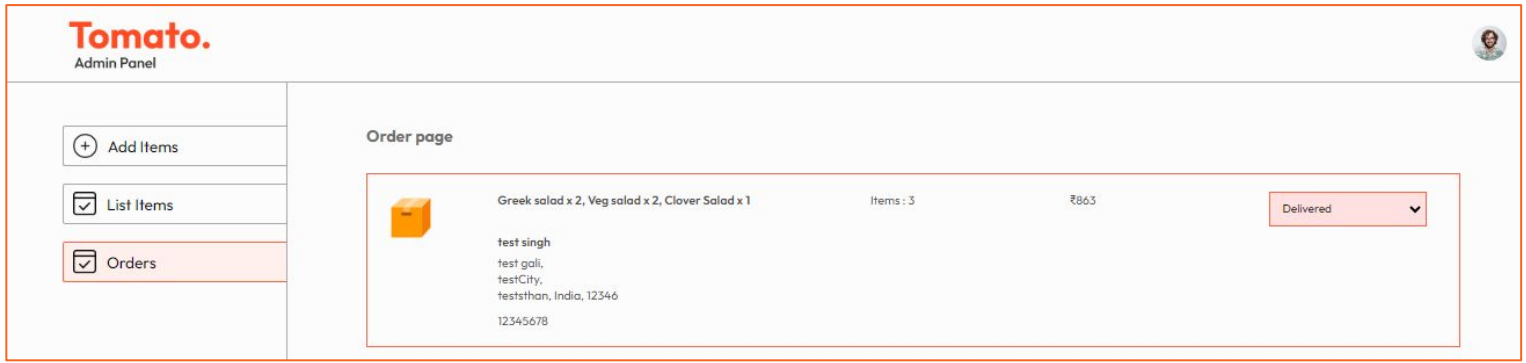
Tomato.
Admin Panel

Add Food Items



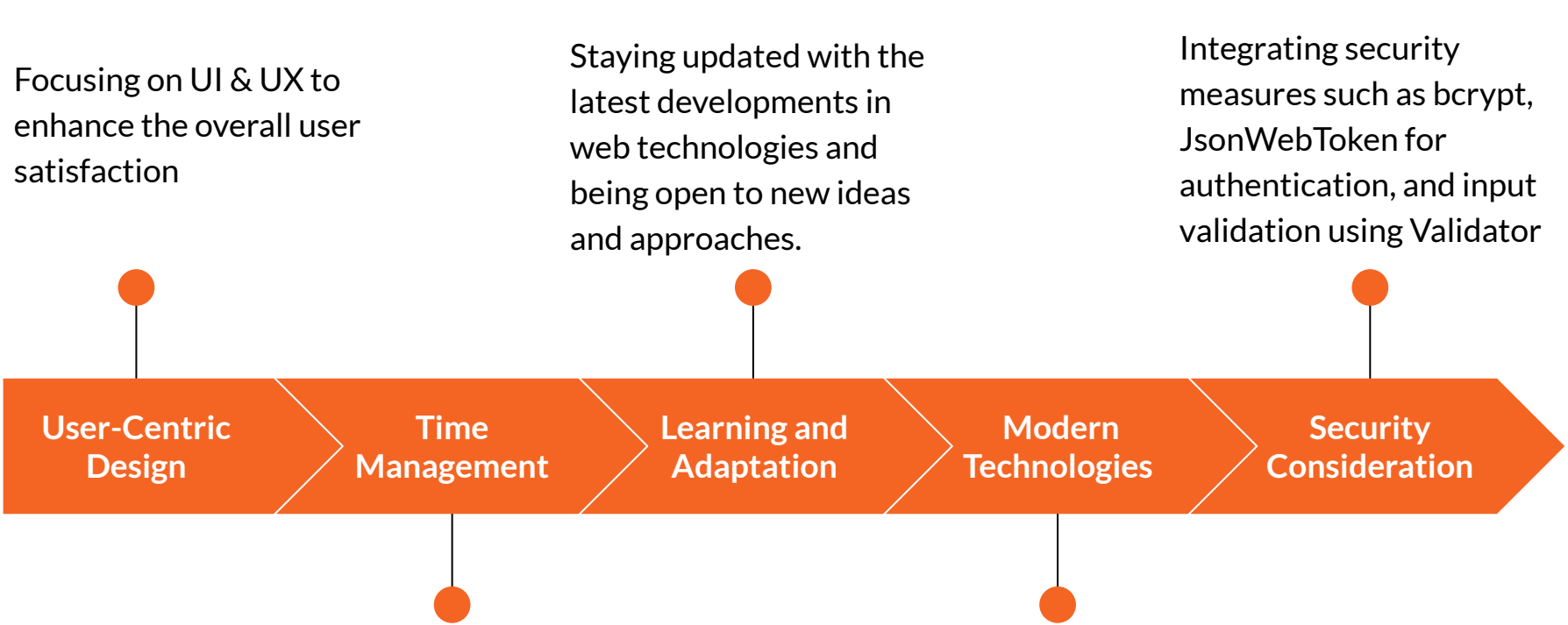
Tomato.
Admin Panel

Items List



Orders Page

Lessons Learned


Focusing on UI & UX to enhance the overall user satisfaction
Staying updated with the latest developments in web technologies and being open to new ideas and approaches.
Integrating security measures such as bcrypt, JsonWebToken for authentication, and input validation using Validator
User-Centric Design
Time Management
Learning and Adaptation
Modern Technologies
Security Consideration
Effective time management was crucial in ensuring that project milestones were met and deadlines were adhered to.
Modern technologies such as the Vite React framework, MongoDB Atlas, and Node.js facilitated the development of a robust and scalable application.

Tomato.

Thank You.

GitHub link - https://github.com/tailormayank/Tomato

Questions ?

Project link - https://tomato-frontend-r1pc.onrender.com/

Admin Panel link - Project link - https://tomato-admin-uelq.onrender.com