

SUSHI GRR

Vamshi Krishna Reddy Purumandla

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

April 2023



The signatures of the individuals below indicate that they have read and approved the project of Vamshi Krishna Reddy Purumandla in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

Dr. Rajvardhan Patil Ph.D., Project Advisor 04/27/2023

Dr. Robert Adams Ph.D., Graduate Program Director 04/27/2023

Dr. Paul M. Leidig Ph.D., Unit head 04/27/2023

ABSTRACT

SUSHI GRR is a Cloud Kitchen project built using modern web development technologies, including React Redux, Axios, Next.js, MongoDB, and REST API.

Customers can browse the kitchen's menu, select items, and add them to their cart. They can then proceed to checkout and pay using either PayPal or cash on delivery. In addition, the application allows customers to view and track the status of their current orders.

React Redux framework is used for state management, making handling complex data flows and managing UI updates easy. It ensures that the application is highly responsive and scalable. Axios fetches data from the MongoDB database, allowing the application to retrieve menu items and order status information quickly and efficiently.

The project also incorporates REST API for seamless front and back-end communication enabling the application to handle complex queries, fetch data from multiple sources, and manage user authentication and authorization. The use of cookie authentication for admin authentication, allowing them to securely log in and access the order history, order status, and other account details. The application also features an admin dashboard, allowing the kitchen staff to manage orders, menu items, and managing and tracking orders.

Overall, this cloud kitchen project provides a powerful and flexible React-based framework that allows customers to order food online conveniently and is an ideal choice for restaurants looking to offer flexible payment options and online ordering to their customers.

Table of Content

Abstract	i
List of Figures	ii
1. Introduction	1
1.1 Introduction	1
1.2 Project Scope	2
1.3 Project Purpose	3
1.4 Design Phase	4
2. Application Structure	7
3. Technologies Used	11
3.1 Next.js	11
3.2 Mongo DB	11
3.3 REST API	13
3.4 Axios	13
3.5 React – Redux	14
3.6 Cookie Authentication	14
4. Conclusion	15
5. Future Work	16
6. References	17

List of Figures

FIG NO.	FIGURE TITLE	PAGE NUMBER
1.1	Customer Use case Diagram	4
1.2	Admin Use case Diagram	4
1.3	Activity Diagram	5
1.4	Sequence Diagram	5
1.5	System Function Diagram	6
2.1	Home Page	7
2.2	Product Page	7
2.3	Cart Page	8
2.4	Order Tracking Page	9
2.5	Admin Dashboard Page	9
2.6	Order Progress	10
2.7	Admin Add Products Page	10
3.1	DB view of Unique Order_ID	12
3.2	REST API	13
3.3	Cookie Authentication	14

CHAPTER 1 – INTRODUCTION

1.1 INTRODUCTION

The online food ordering industry has seen tremendous growth in recent years, with more and more people prefer ordering food online. In this report, I present a Cloud Kitchen website that allows customers to order food online and track their order status. The website also provides an admin dashboard for managing products and updating order status, ensuring that customers are constantly updated with their order status.

The website was developed to provide a seamless and user-friendly experience for customers. Customers can browse the menu, select items, and place orders easily using the website's intuitive interface. They can also track the status of their order in real time, ensuring that they are always aware of when their food will arrive.

The website also provides an admin dashboard for managing products and orders. Admins can easily add new products to the menu and manage existing products using the dashboard. They can also update the status of orders, ensuring that customers are always informed of their order status.

The website was built using modern web technologies React Next.js, and MongoDB. The front end was developed using React, a popular JavaScript library for building user interfaces, and was used as the database, providing a scalable and flexible data storage solution.

The website was designed to be responsive and mobile-friendly, ensuring that customers can order food from anywhere, using any device. The website was also built with security in mind, with all sensitive data being encrypted and stored securely.

1.2 PROJECT SCOPE

This project's scope was to develop an online food ordering website that allows customers to order food online and track their order status while providing an admin dashboard for managing products and updating order status.

The website was developed using modern web technologies such as React, Node.js, and MongoDB. The front end was designed to be responsive and mobile-friendly, allowing customers to place orders from anywhere, using any device. The website was also designed with security, with all sensitive data being encrypted and stored securely.

The website's admin dashboard allows admins to manage products, view orders, and update order status. The dashboard was designed to be user-friendly and easy to navigate, allowing admins to manage products and orders efficiently.

The scope of the project included the following key features:

- Menu browsing and item selection
- Order placement and tracking
- Admin dashboard for managing products and orders
- Real-time order status updates
- Payment processing and order confirmation

The project scope did not include the development of a mobile application, integrating with third-party delivery services, or implementing advanced analytics or reporting features.

Overall, this project's scope was focused on providing a seamless and user-friendly experience for customers while also providing a robust and efficient management platform for admins.

1.3 PROJECT PURPOSE

This project aimed to develop an online food ordering website to help customers order food online and track order status while providing an admin dashboard for managing products and updating order status.

The website was designed to provide customers with a seamless and user-friendly experience, allowing them to quickly browse the menu, select items, and place orders. Customers can also track their orders in real-time, giving them control and visibility throughout the order fulfillment process.

The admin dashboard was designed to be user-friendly and efficient, allowing admins to manage products and orders efficiently. In addition, the dashboard provides an easy-to-use interface for managing products, viewing orders, and updating order status, ensuring that customers are always informed about the status of their orders.

This project aimed to provide a scalable and secure platform that can handle large volumes of data and user requests. The website was developed using modern web technologies such as React, Node.js, and MongoDB, providing a robust and efficient platform for businesses looking to offer online food ordering services.

1.4 DESIGN PHASE

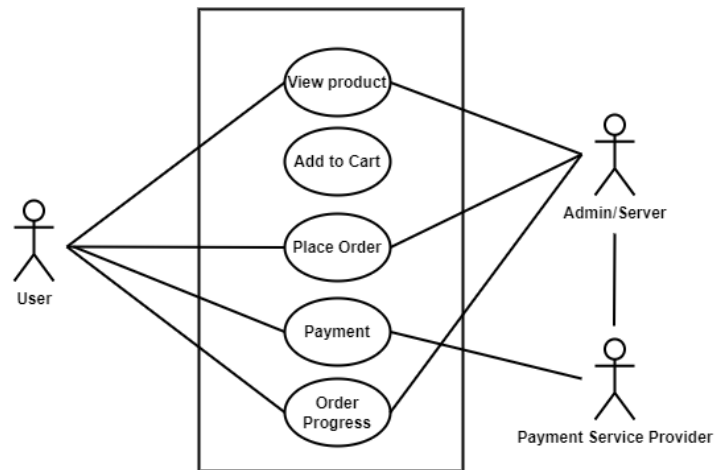


Figure 1.1: Customer Use Case Diagram

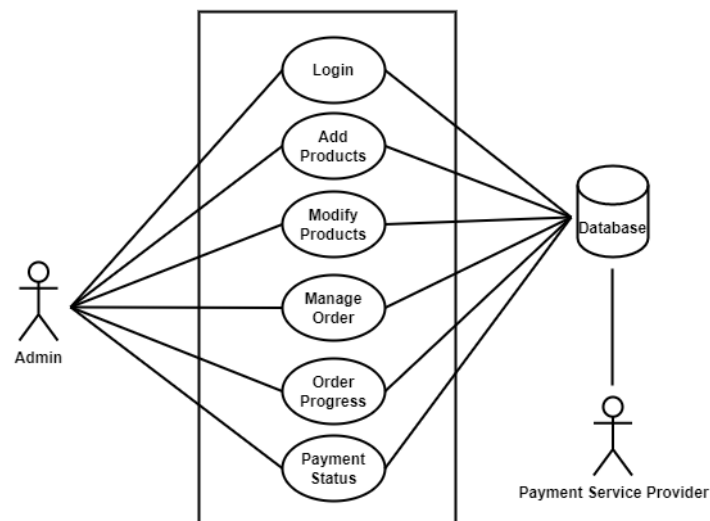


Figure 1.2: Admin Use Case Diagram

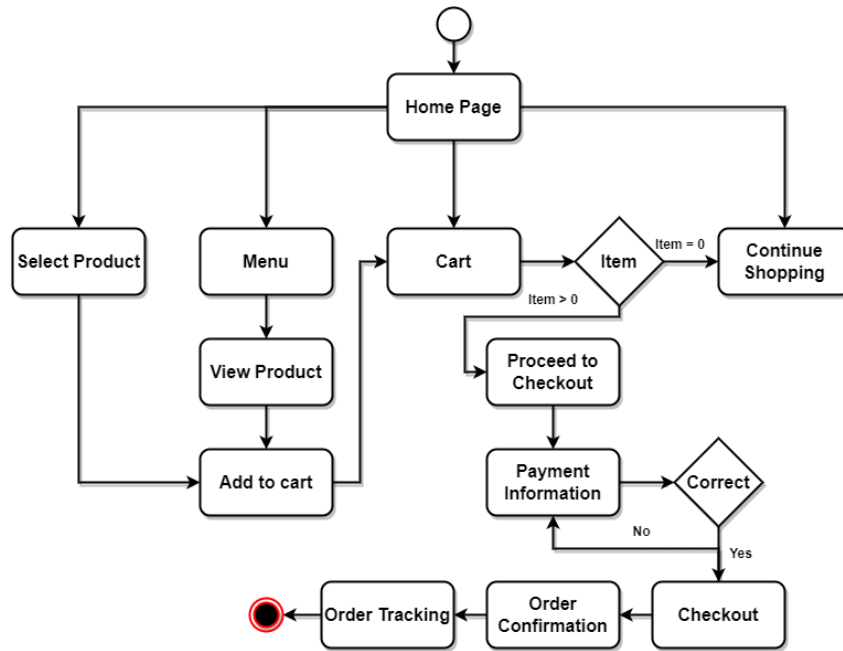


Figure 1.3: Activity Diagram

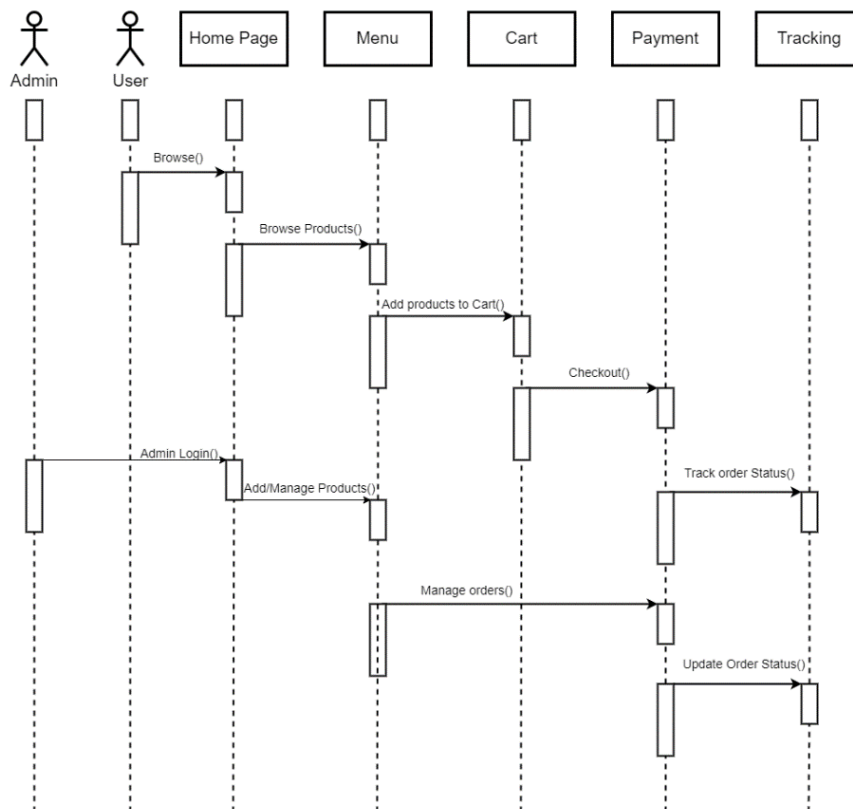


Figure 1.4: Sequence Diagram

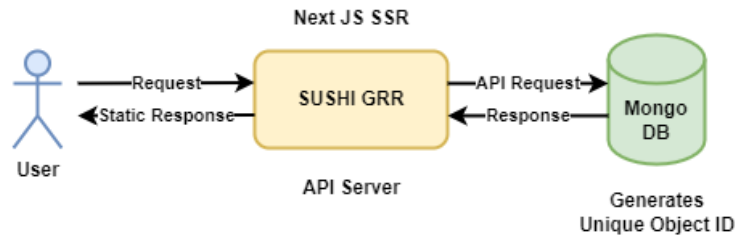


Figure 1.5: System Function Diagram

Server-side rendering (SSR) is a popular technique for rendering a client-side single-page application (SPA) on the server and then sending a fully rendered page to the client. This allows for dynamic components to be served as static HTML markup.

The HTML is rendered immediately with all the static elements. In the meantime, the browser downloads and executes the JS code, after which the page becomes interactive. The browser on the client side now handles the interactions on this page. The browser sends a request to the server through APIs for any new content or data, and only the newly required information is fetched. This approach can be helpful in search engine optimization (SEO) when indexing does not handle JavaScript properly. It may also be beneficial when downloading a large JavaScript bundle is impaired by a slow network.

CHAPTER 2 – APPLICATION STRUCTURE

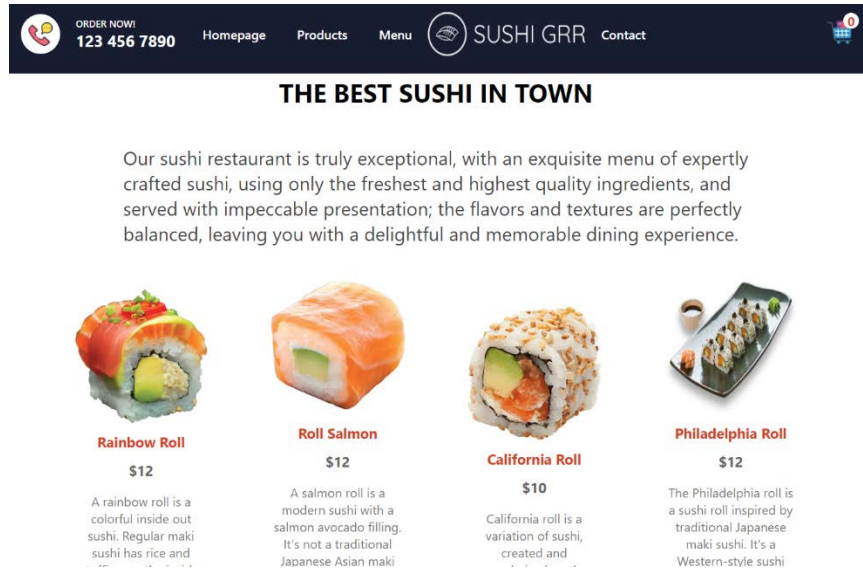


Figure 2.1: Home Page

The homepage provides an overview of the website's offerings, with a navigation menu allowing customers to browse and view their cart. The homepage also features a banner image highlighting promotions or featured menu items, providing an engaging visual element to the website.

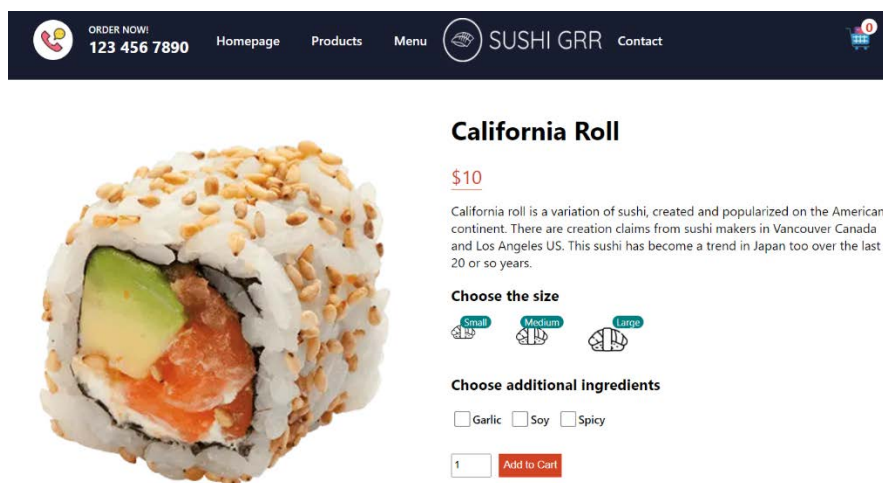










Figure 2.2: Product page

The product page is designed to be visually appealing and informative, providing customers with all the details they need to make an informed decision about the product. In addition, the page features a high-quality image of the product, along with a detailed description. The product page also includes the price of the product and an option to select the quantity of the product.

<div><div> ORDER NOW! 123 456 7890</div><div>Homepage</div><div>Products</div><div>Menu</div><div> SUSHI GRR</div><div>Contact</div><div></div></div>						
Product	Name	Extras	Price	Quantity	Total	
	California Roll	Soy,	\$18	1	\$18	
	Philadelphia Roll	Soy sauce,	\$15	1	\$15	
	Nigiri Sushi	Ginger sauce,	\$17	1	\$17	
	Roll Salmon	Ginger sauce,	\$16	1	\$16	
	California Roll		\$10	1	\$10	

CART TOTAL

Subtotal: \$76

Discount: \$0.00

Total: \$76

CASH ON DELIVERY

PayPal

Figure 2.3: Cart Page

The cart page allows customers to review their orders. The page also includes a summary of the total cost of the order, including any applicable taxes or fees. Customers can select a payment method and confirm their order.

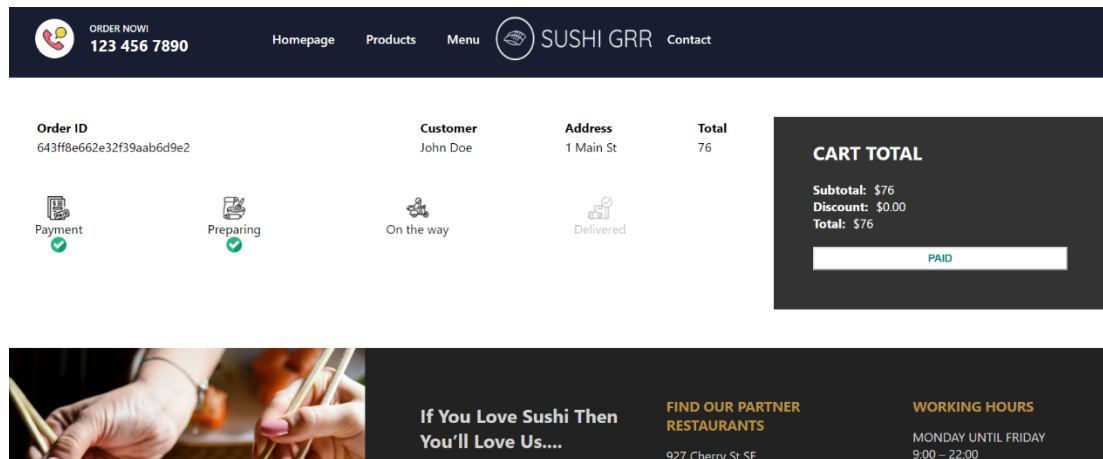


Figure 2.4: Order Tracking Page

The order tracking page allows customers to track the status of their order in real time, providing updates on order preparation, shipping, and delivery. In addition, the page includes a progress indicator to help customers keep track of their orders.












<div>  <div>ORDER NOW!</div> <div>123 456 7890</div> </div> <div> Homepage Products Menu <div>  <div>SUSHI GRR</div> </div> Contact </div>				
Product				
Image	Id	Title	Price	Action
	64147...	Rainbow Roll	\$12	Edit Delete
	64149...	Roll Salmon	\$12	Edit Delete
	64271...	California Roll	\$10	Edit Delete
	643fe...	Philadelphia Roll	\$12	Edit Delete
	643fe...	Roll Perch Salmon	\$12	Edit Delete
	643fe...	Nigiri Sushi	\$12	Edit Delete
	643fe...	Nigiri Tuna	\$12	Edit Delete


Figure 2.5: Admin Dashboard Page

	643fe...	Nigiri Tuna	\$12	Edit Delete
	643fe...	Ebi Nigiri	\$12	Edit Delete

Orders

Id	Customer	Total	Payment	Status	Action
64268...	Adam Jr	\$12	paid	Delivered!	Next Stage
64272...	vamshi	\$0	cash	Delivered!	Next Stage
64273...	Adam Jr	\$18	paid	Delivered!	Next Stage
643ff...	John Doe	\$76	paid	on the way	Next Stage

Figure 2.6: Order Progress



Add a new Sushi Item

Choose an image

No file chosen

Title

Desc

Prices

Small
 Medium
 Large

Extra

Item
 Price

Figure 2.7: Admin Add Products Page

The website's admin dashboard provides an easy-to-use interface for managing products and orders. The dashboard includes tabs for managing products, viewing orders, and updating order status. The product management tab allows admins to add new products, edit existing products, and delete products as needed. The orders tab allows admins to view all current and past orders, and the order status tab provides a convenient way to update the status of orders as they progress through the fulfillment process.

CHAPTER 3 – TECHNOLOGIES USED

3.1 NEXT.JS

Next.js is a popular React framework used to develop the front of the Sushi GRR cloud kitchen application. Next.js provides several benefits, making it an ideal choice for building high-performance web applications like this. It provides a server-side rendering (SSR) capability, which enables the web application to load faster and provide a smoother user experience. Second, SSR benefits web applications like Sushi GRR, which have much dynamic content that needs to be loaded quickly.

Next.js provides automatic code splitting and optimized image loading capabilities, ensuring the application loads quickly and efficiently, even on slower internet connections. This is important for a cloud kitchen application like ours, where users expect a fast and seamless experience. It is highly flexible and customizable, allowing developers to integrate additional features and functionality into the application easily. This flexibility makes it easy to adapt Sushi GRR to changing user needs and to scale the application as needed to handle increasing traffic and demand.

3.2 MONGO DB

MongoDB is a robust NoSQL database used to develop the application's backend. However, MongoDB provides several benefits, making it an ideal choice for building high-performance, scalable web applications.

One of the primary benefits of MongoDB is its flexible data model, which allows for the storage and management of complex data structures. This is particularly useful for an application like ours, which requires the storage of customer profiles, order histories, menu items, and other data.

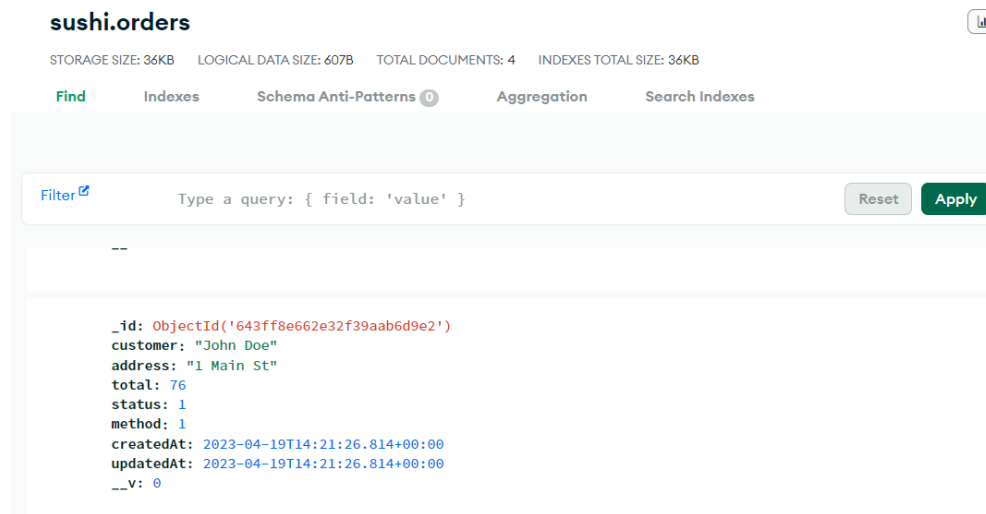


Figure 3.1: DB view of Unique Order_ID assigned to new orders.

Another benefit of MongoDB is its ability to handle high volumes of data and traffic. MongoDB is designed to be highly scalable, which means that it can easily handle large amounts of data and user requests without compromising performance. In addition, MongoDB provides built-in replication and sharding capabilities, which help to ensure high availability and fault tolerance. This means that even if one database node goes down, the application can continue to function without interruption.

3.3 REST API

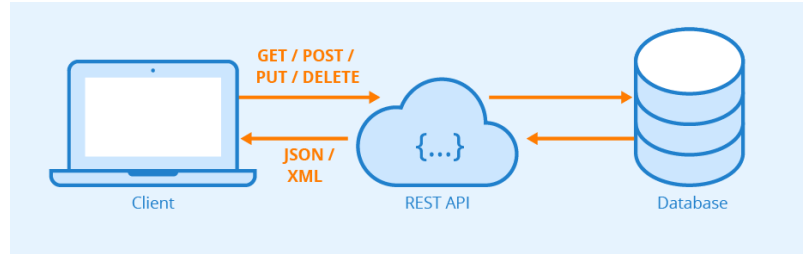


Figure 3.2: REST API - Author: Seobility - License: [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST, and DELETE data types, which refers to the reading, updating, creating, and deleting operations concerning resources. An API for a website is code that allows two software programs to communicate with each other. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

RESTful API also called a RESTful web service or REST API, is based on representational state transfer (REST), an architectural style and approach to communications often used in web services development. REST technology is generally preferred over other similar technologies. REST uses less bandwidth, making it more suitable for efficient internet usage. RESTful APIs can also be built with programming languages such as JavaScript or Python.

3.4 AXIOS

Axios is a lightweight HTTP client based on the \$http service within Angular.js v1.x and is similar to the native JavaScript Fetch API. Axios is promise-based, allowing us to take advantage of JavaScript's async and await for more readable asynchronous code.

We can also intercept and cancel requests, and there is built-in client-side protection against cross-site request forgery.

3.5 REACT - REDUX

React Redux is the official React binding for Redux. It allows React components to read data from a Redux Store and dispatch Actions to the Store to update data. Redux helps apps to scale by providing a sensible way to manage the state through a unidirectional data flow model. React Redux is conceptually simple. It subscribes to the Redux store, checks to see if the data our component wants to be changed, and re-renders our component.

Flux inspired Redux, who studied Flux architecture and omitted unnecessary complexity.

- Redux does not have a Dispatcher concept.
- Redux has only one Store, whereas Flux has many Stores.
- The Action objects will be received and handled directly by Store.

3.6 COOKIE AUTHENTICATION

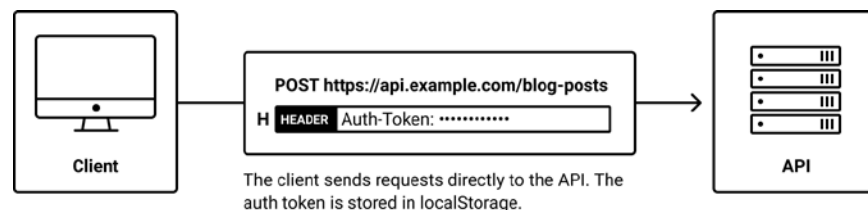


Figure 3.3: Cookie Authentication

Cookie authentication is a secure and popular method used in web applications for user authentication, including the admin login for the online food ordering website. It creates a unique session ID when a user logs in and stores it as a cookie in the user's browser. The session ID is passed with each request, and the server uses it to authenticate the request and verify the user's identity. This ensures that only authenticated users can access the admin dashboard and improves the user experience by reducing the need for repeated logins.

CHAPTER 4 – CONCLUSION

In conclusion, the SUSHI GRR website developed in this project is a user-friendly and secure platform providing a seamless experience for customers and admins. Its intuitive interface allows customers to browse the menu, place orders, and track them in real-time. In addition, the website's admin dashboard provides an easy way for admins to manage products and orders, ensuring that customers are always informed about the status of their orders.

The use of modern web technologies such as React, Next.js, and MongoDB provides a scalable and flexible platform that can handle large volumes of data and user requests. The website is mobile-responsive, ensuring customers can access it from any device, anywhere, and anytime.

Security is a top priority, with all sensitive data being encrypted and stored securely. In addition, the website can be hosted on a secure server, ensuring that all user data is protected from potential security breaches.

Overall, the cloud kitchen website presented in this project report is an excellent example of how modern web technologies can be used to develop a robust and user-friendly platform that meets the needs of both customers and admins. With its mobile responsiveness, security, and scalability, the website is poised to become a go-to platform for businesses looking to offer online food ordering services.

CHAPTER 5 – FUTURE WORKS

Several future works can be considered for this project report's online food ordering website. One possible future work is implementing a recommendation system that suggests menu items to customers based on their previous orders or search history.

Another potential improvement would be integrating the website with third-party delivery services such as Uber Eats or Grub Hub, allowing customers to choose from a broader range of delivery options. In addition, the website can be enhanced with additional payment options, such as Apple Pay or Google Pay, to provide customers with more convenient payment options.

Finally, the website's admin dashboard can be improved with additional features, such as analytics and reporting, to give admins insights into customer behavior and order patterns. These features can help admins make informed decisions about product offerings and marketing strategies. Several potential future works can enhance the website's functionality and user experience, making it an even more valuable tool for businesses looking to offer online food ordering services.

CHAPTER 6 - REFERENCE

[1] Riddy, Jannat & Rajee, Alimul & prokashpal752@gmail.com, & abdulhashem109904@gmail.com,. (2022).
ONLINE FOOD ORDERING SYSTEM.

[2] <https://auth0.com/blog/ultimate-guide-nextjs-authentication-auth0/>

[3] <https://www.mongodb.com/docs/>

[4] <https://nextjs.org/docs>

[5] <https://react.dev/learn>

[6] <https://www.digitalocean.com/community/tutorials/react-server-side-rendering>

[7] <https://www.javatpoint.com/react-redux>

[8] <https://www.digitalocean.com/community/tutorials/react-axios-react>

[9] <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>

[10] <https://paypal.github.io/react-paypal-js/?path=/story/example-paypalbuttons--default>

[11] <https://maxschmitt.me/posts/next-js-http-only-cookie-auth-tokens>

[12] <https://www.tutorialspoint.com/a-guide-to-server-side-rendering-in-react>

[13] <https://medium.com/geekculture/designing-complex-rest-api-structure-with-state-machine-cc2b8804d467>

[14] <https://www.cloudsigma.com/using-http-client-axios-in-a-react-application-a-tutorial>