**Table Of Contents**

Introduction …………………………………………………………………………………………………………………………………………2

Background …………………………………………………………………………………………………………………………………..2

Purpose ………………………………………………………………………………………………………………………………………..2

Abstract …………………………………………………………………………………………………………………………………………2

Requirements ………………………………………………………………………………………………………………………………………2

Functional Requirements ………………………………………………………………………………………………………………3-4

Non-Functional Requirements ……………………………………………………………………………………………………….4-5

Software Engineering Model ………………………………………………………………………………………………………………….5

Evaluation And Selection of Software Engineering Model ……………………………….……………………………...5

Selection of Agile Methodology …………………………………………………………………………………………………………….6

Implementation Detail …………………………………………………………………………………………………………………………..6

Front-End Development …………………………………………………………………………………………………………………6

Back-End Development ………………………………………………………………………………………………………………….6

Data-Base Management …………………………………………………………………………………………………………………6

User Interface ………………………………………………………………………………………………………………………………………6-7

Visual Design …………………………………………………………………………………………………………………………………7

Navigation And Interaction ……………………………………………………………………………………………………………7

Testing And Quality Assurance ……………………………………………………………………………………………………………..7

Testing Approach …………………………………………………………………………………………………………………………..7

Testing Cases and Execution …………………………………………………………………………………………………………7

Quality Assurance …………………………………………………………………………………………………………………………7

Deployment And Maintenance ………………………………………………………………………………………………………….7-8

Deployment Process ……………………………………………………………………………………………………………………8

Maintenance And Updates ………………………………………………………………………………………………………….8

Conclusion ………………………………………………………………………………………………………………………………………….8

Model Diagram ……………………………………………………………………………………………………………………………………9

Flow Diagram ………………………………………………………………………………………………………………………………………10

Entity-Relationship Diagram …………………………………………………………………………………………………………………11

System Prototype …………………………………………………………………………………………………………………………………12

References ……………………………………………………………………………………………………………………………………………13

1. **Introduction**
   1. **Background:**

The Food Hut website aims to revolutionize the food industry by providing a convenient platform for users to order delicious meals from their favourite restaurants. With the increasing popularity of online food ordering, there is a growing demand for a user-friendly and efficient system that connects restaurants, customers, and delivery services. The Food Hut website seeks to address this need by offering a seamless and enjoyable food ordering experience.

The food industry has seen a significant shift towards digital platforms in recent years, with customers increasingly relying on online ordering and delivery services. This trend has been accelerated by advancements in technology, the ubiquity of smartphones, and changing consumer preferences. By leveraging these opportunities, the Food Hut website aims to tap into the thriving market for online food delivery.

* 1. **Purpose:**

The purpose of this report is to document the development process of the Food Hut website, providing an in-depth analysis of the various aspects involved in its creation. It will outline the project goals, scope, methodologies, and implementation details. Additionally, it will discuss the challenges faced, lessons learned, and the overall impact of the website on the food industry.

This report serves as a comprehensive guide for stakeholders, developers, and future researchers interested in understanding the development and implementation of an online food ordering platform. It aims to provide insights into the technical, design, and business considerations involved in creating such a website.

* 1. **Abstract:**

The abstract provides a concise summary of the report, capturing the key elements and findings of the project. It highlights the objectives, methodologies, and outcomes of the development process. The abstract briefly mentions the selection of the Agile software engineering model, the implementation of front-end and back-end technologies, the design of the user interface, the testing and quality assurance measures, and the deployment and maintenance process. The abstract concludes by emphasizing the significance of the Food Hut website in meeting the needs of customers and revolutionizing the food industry.

1. **Requirements**

**2.1 Functional Requirements:**

* **User Registration and Authentication:** Allow users to create accounts and authenticate themselves for personalized experiences and secure transactions.
* **Browse and Search**: Provide a user-friendly interface to browse and search for various food options, including restaurants, recipes, or specific dishes.
* **Menu and Ordering:** Display menus with detailed information about dishes, including descriptions, prices, and dietary options. Enable users to place orders and customize their selections.
* **Online Payments:** Integrate secure online payment options to facilitate seamless transactions for food orders.
* **Delivery and Pickup** **Options:** Provide users with the choice to have their food delivered to a specified address or opt for pickup from the restaurant.
* **Ratings and Reviews:** Allow users to rate and review restaurants, dishes, and overall food experiences to help other users make informed choices.
* **Personalized Recommendations:** Utilize user preferences, order history, and machine learning algorithms to offer personalized food recommendations.
* **Social Sharing and Integration:** Enable users to share their food experiences on social media platforms and integrate with popular social media channels.
* **Loyalty and Rewards: Implement** loyalty programs or reward systems to incentivize user engagement and encourage repeat orders.
* **Customer Support**: Provide customer support channels, such as in-app messaging or live chat, to address user queries or issues.
* **Restaurant Management:** Provide an interface for restaurants to manage their menus, update dish availability, and track orders.
* **Admin Dashboard:** Develop an admin dashboard to monitor and manage user accounts, reviews, and overall app performance.
* **Geolocation:** Incorporate geolocation functionality to allow users to find nearby restaurants, delivery services, or pickup options.
* **Order Tracking**: Enable users to track the status of their orders in real-time, including updates on preparation, packaging, and delivery.
* **Special Dietary Needs:** Provide options for users to filter and search for specific dietary requirements, such as vegetarian, vegan, gluten-free, or allergen-free options.
* **Favourites and Saved Orders:** Allow users to save their favourite restaurants or dishes for quick access in the future. Enable users to reorder their previous orders easily.
* **Push Notifications:** Send push notifications to users for order updates, special offers, or personalized recommendations.
* **Social Login and Sharing**: Allow users to sign in using their social media accounts for a seamless login experience. Enable users to share their food orders or restaurant experiences on social media platforms.
* **Restaurant Ratings and Certifications:** Display restaurant ratings, certifications, or hygiene scores to help users make informed decisions about the places they order from.
* **Multi-Language Support:** Provide support for multiple languages to cater to a diverse user base.
* **Pre-order and Scheduled Delivery:** Allow users to place pre-orders for a specific time or schedule their food delivery in advance.
* **Advanced Search Filters:** Implement advanced search filters, such as price range, cuisine type, delivery time, or restaurant ratings, to help users narrow down their choices.
* **In-App Chat or Call Support:** Enable users to contact customer support via in-app chat or initiate calls directly from the app for immediate assistance.
* **Integration with Food Delivery Platforms**: Integrate with popular food delivery platforms or APIs to expand restaurant options and delivery coverage.
* **User Profile and Order History:** Provide users with a profile section to manage their account information and view their order history.

**2.2 Non-functional Requirements:**

The non-functional requirements of the Food Hut website are as follows:

1. **Performance:**

* The website should have fast loading times to ensure a seamless user experience.
* It should be able to handle a large number of concurrent users without significant slowdowns.
* Response times for essential operations should be within acceptable limits.

1. **Security:**

* The website should implement robust security measures to protect user data and prevent unauthorized access.
* Secure communication protocols should be used to encrypt sensitive information.
* Common security vulnerabilities should be mitigated.

1. **Usability:**

* The website should have an intuitive and user-friendly interface.
* Content should be organized logically for easy navigation.
* Clear instructions should be provided for users to complete tasks efficiently.

1. **Scalability:**

* The website should be designed to handle increased user traffic without performance degradation.
* It should be able to scale horizontally by adding more servers if necessary.

1. **Compatibility:**
   * The website should be compatible with popular web browsers such as Chrome, Firefox, Safari, and Edge.
   * It should work well on different operating systems and screen sizes.
2. **Accessibility:**

* The website should adhere to accessibility standards to accommodate users with disabilities.
* It should provide alternatives for visual and auditory elements.
* Proper colour contrast and font sizes should be implemented.

These non-functional requirements ensure that the Food Hut website performs well, provides a secure environment, is user-friendly, can handle increased demand, is compatible with various platforms, and is accessible to a wide range of users.

1. **Software Engineering Model Selection:**
   1. **Evaluation of Software Engineering Models:**

This section provides an evaluation of different software engineering models and methodologies based on their suitability for the development of the Food Hut website. It compares and contrasts models such as Waterfall, Agile, Spiral, and Lean, considering factors such as project scope, flexibility, iterative development, stakeholder involvement, and risk management. The advantages and disadvantages of each model are discussed, highlighting their applicability to the specific requirements and constraints of the project.

* 1. **Selection of Agile Model:**

After careful evaluation, the Agile software engineering model is chosen as the most suitable approach for the development of the Food Hut website. The section explains the reasons behind this selection, such as the need for flexibility, continuous feedback, incremental development, and close collaboration with stakeholders. It discusses the key principles and practices of Agile, including the use of sprints, user stories, daily stand-ups, and retrospectives. The benefits of adopting an Agile approach, such as faster time to market, adaptability to changing requirements, and improved customer satisfaction, are emphasized.

1. **Implementation Details:**
   1. **Front-end Development:**

This section delves into the implementation details of the front-end development process for the Food Hut website. It discusses the technologies and frameworks used, such as HTML, CSS, and JavaScript, as well as any front-end libraries or frameworks like React or Angular. The section covers the overall architecture of the front-end, including component-based design and the use of responsive layouts to ensure optimal user experience across different devices. It also highlights any specific design patterns or best practices followed during the development.

* 1. **Back-end Development:**

The back-end development details are explained in this section. It focuses on the server-side technologies and frameworks used, such as Node.js, Express.js, and any relevant database management systems like MySQL or MongoDB. The section outlines the architecture of the back-end, including the RESTful API design, routing, and data persistence strategies. It also discusses any third-party integrations or external services utilized, such as payment gateways or SMS notifications.

* 1. **Database Management:**

The database management aspect of the Food Hut website is covered in this section. It describes the database technologies chosen, the data modelling process, and the schema design. It explains how the database is structured to store essential information, such as user profiles, menus, orders, and feedback. It also addresses data security measures, such as encryption, access control, and data backup strategies.

1. **User Interface Design:**
   1. **Visual Design:**

This section focuses on the visual design aspects of the Food Hut website. It discusses the colour scheme, typography, and overall visual aesthetics chosen to create an appealing and cohesive user interface. It may include the use of mock-up tools or graphic design software to create wireframes and prototypes, providing a visual representation of the website's layout and elements. The section also emphasizes the importance of maintaining brand consistency and user-centred design principles throughout the interface.

* 1. **Navigation and Interaction:**

The navigation and interaction design of the Food Hut website are explained in this section. It covers the design of the website's menus, navigation bars, search functionalities, and interactive elements. It discusses the user flow and the design decisions made to ensure smooth and intuitive navigation. The section may also touch upon responsive design considerations to optimize the user experience on different screen sizes and devices.

1. **Testing and Quality Assurance:**
   1. **Testing Approach:**

This section details the testing approach adopted for the Food Hut website. It discusses the types of testing performed, such as unit testing, integration testing, system testing, and acceptance testing. It explains the use of testing frameworks or tools, such as Jest or Selenium, and the establishment of test environments. It may also mention the use of test-driven development (TDD) or behaviour-driven development (BDD) techniques.

* 1. **Test Cases and Execution:**

The test cases and execution process for the Food Hut website are documented in this section. It provides examples of test cases for different functionalities, highlighting the expected outcomes and any specific edge cases considered. It explains the test execution process, including test scripts, test data, and any automated testing procedures implemented. The section also discusses the identification and resolution of defects or issues encountered during testing.

* 1. **Quality Assurance Measures:**

This section outlines the quality assurance measures taken during the development of the Food Hut website. It discusses code reviews, peer testing, and adherence to coding standards or style guides. It may mention the use of continuous integration and continuous deployment (CI/CD) practices to ensure a high level of software quality. The section also addresses security testing, performance testing, and any accessibility testing conducted to meet industry standards and user expectations.

1. **Deployment and Maintenance:**
   1. **Deployment Process:**

This section explains the deployment process for the Food Hut website, including the selection of hosting providers, configuration of servers, and deployment strategies. It discusses the use of containerization technologies like Docker and the deployment of the website on cloud platforms like AWS or Azure. It may also touch upon scalability considerations and the use of load balancing or auto-scaling techniques.

* 1. **Maintenance and Updates:**

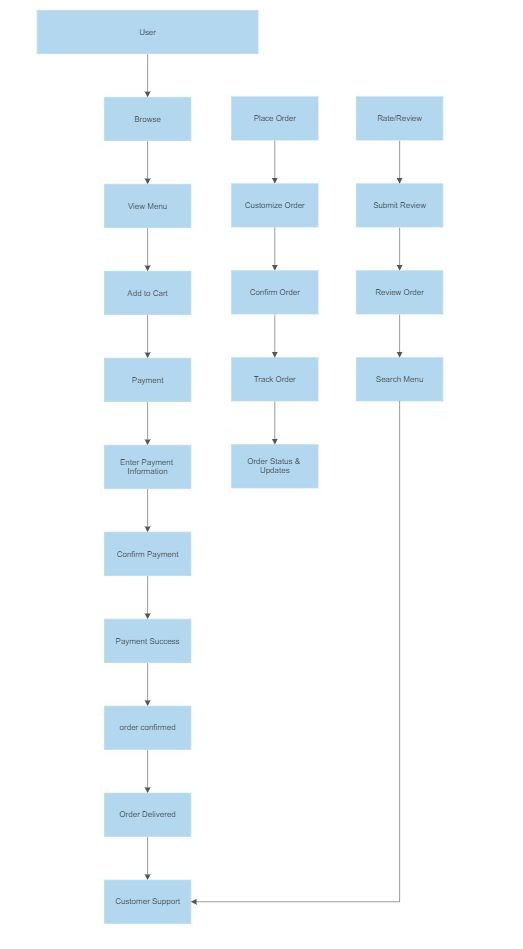
The maintenance and updates aspect of the Food Hut website is covered in this section. It discusses the importance of regular maintenance tasks such as bug fixing, performance optimization, and security updates. It outlines the process of addressing user feedback and implementing feature enhancements or new functionalities. The section also mentions the use of monitoring tools and analytics to track website performance and user behaviour, facilitating continuous improvement.

1. **Conclusion:**

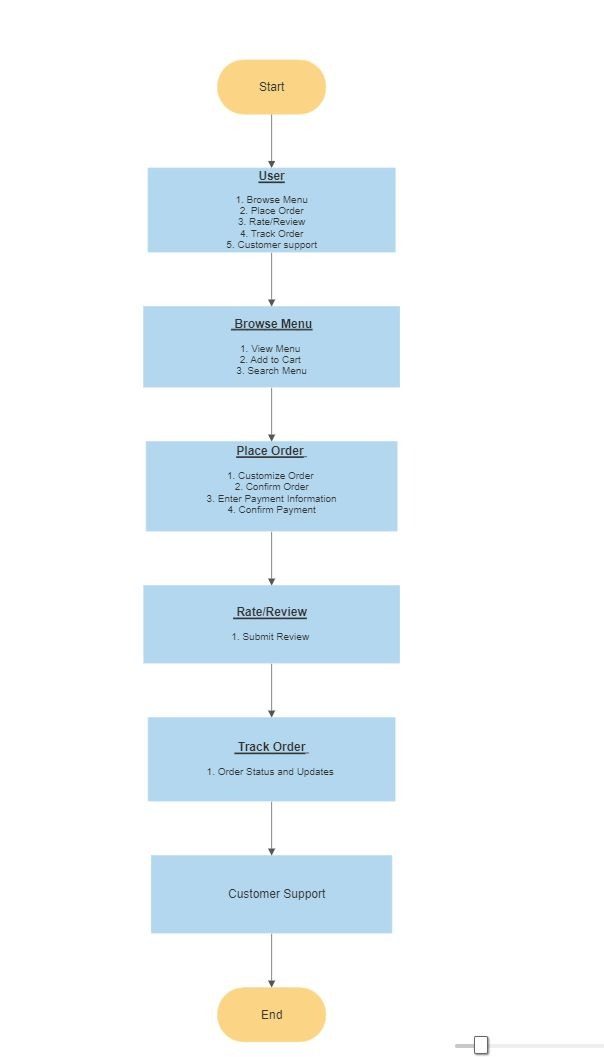
The conclusion section provides a summary of the entire report, highlighting the key findings, achievements, and challenges faced during the development of the Food Hut website. It reflects on the success of the project in meeting its objectives and the potential impact it can have on the food industry. The section also discusses future possibilities for expansion, improvement, and integration with emerging technologies.

1. **Model Diagram:**

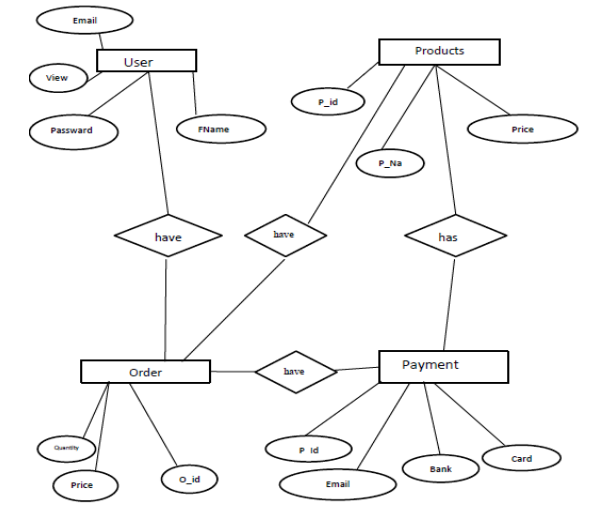
* In this more complex diagram, the main flow of the food app involves various steps and interactions. The user can browse the menu, view details of specific items, add them to the cart, customize the order, and confirm it. They can then proceed to make a payment by entering their payment information.
* After confirming the payment, they receive a payment success message and an order confirmation. Once the order is confirmed, the user can track the order status and receive updates. Upon delivery, they are notified that the order has been delivered. Throughout the process, the user has the option to rate and review their experience. If they require assistance, they can reach out to customer support.



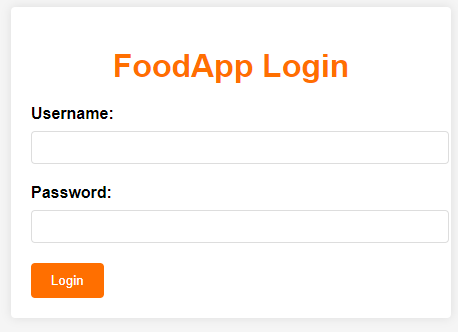
1. **Flow Diagram:**

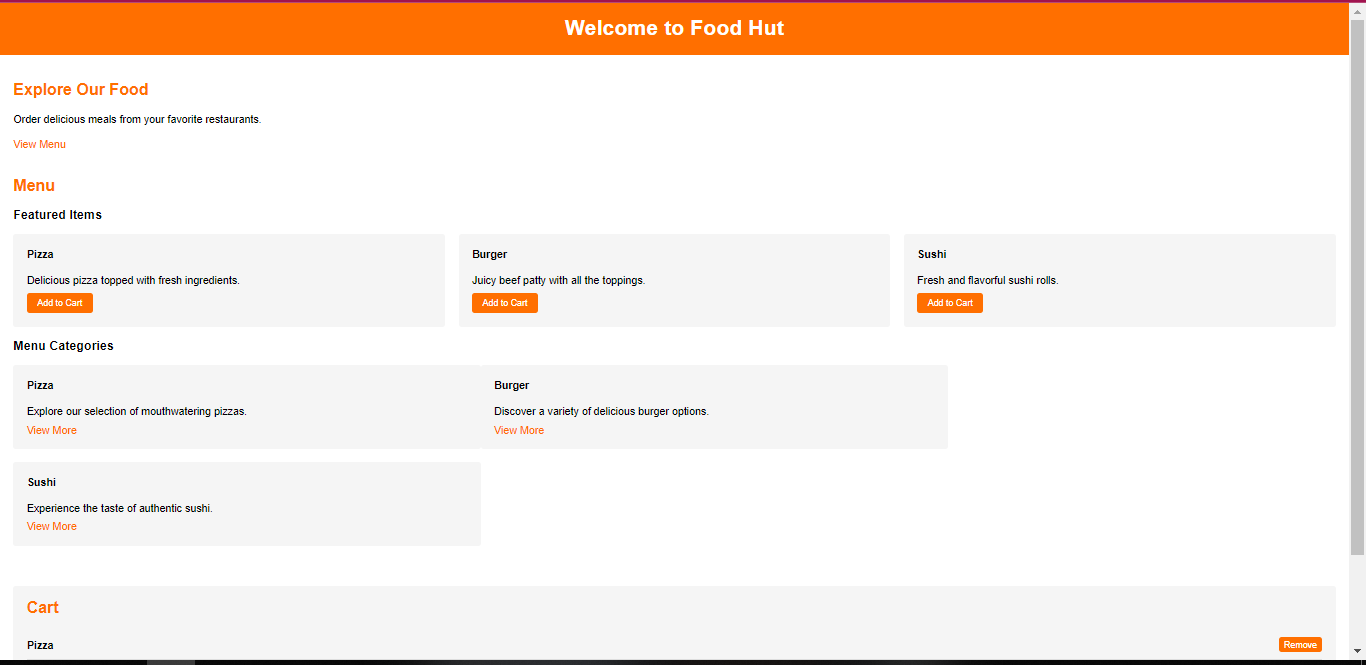


1. **Entity Relationship Model:**



1. **Model Prototype:**





1. **References:**

[1] <https://www.scnsoft.com/blog/software-development-models>

[2] <https://www.freeprojectz.com/project-report/1778>

[3] <https://www.scribd.com/document/557203348/Our-Project-Report-Online-Food-Ordering>

[4] <https://www.slideshare.net/AshwinBicholiya/food-delivery-application-report>

**THE END😊**