



islington college
(इस्लिंग्टन कॉलेज)

Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

25% FYP Interim Report

Semester

2023-24 Autumn

FoodAaja(Food ordering web app with real-time recommendation System)

Student Name:

London Met ID:

College ID:

Internal Supervisor:

External Supervisor:

Assignment Due Date:

Assignment Submission Date:

Word Count (Where Required): 3478

I confirm that I understand my coursework needs to be submitted online via MST under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Sample - BIAI.docx

 Islinton College,Nepal

Document Details

Submission ID

trn:oid:::3618:73855439

124 Pages

Submission Date

Dec 12, 2024, 2:05 PM GMT+5:45

10,895 Words

Download Date

Dec 13, 2024, 8:28 AM GMT+5:45

62,503 Characters

File Name

Sample - BIAI.docx

File Size

73.0 KB

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **105** Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks
-  **2** Missing Quotations 0%
Matches that are still very similar to source material
-  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

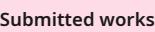
-  105 Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks
-  2 Missing Quotations 0%
Matches that are still very similar to source material
-  0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1		Swinburne University of Technology on 2023-06-05 0%
2		HELP UNIVERSITY on 2013-09-09 0%
3		Southern New Hampshire University - Continuing Education on 2023-07-23 0%
4		University of Maryland, Global Campus on 2022-11-01 0%
5		Asia Pacific University College of Technology and Innovation (UCTI) on 2020-08-19 0%
6		www.nofoodwaste.in 0%
7		Manchester Metropolitan University on 2016-04-15 0%
8		HELP UNIVERSITY on 2018-05-21 0%
9		www.geeksforgeeks.org 0%
10		Adventist University of Central Africa on 2023-08-25 0%

Summary

This interim report details the development of "FoodAaja," a web application designed to enhance the food ordering experience with features like real-time weather-based recommendations, budget-friendly filtering, and order customization. The project, currently in Sprint 1, has successfully implemented server infrastructure, database connections, and customer login features. Key functionalities like restaurant and menu listing are halfway through, with plans to develop payment, checkout, and map integration in future sprints. Utilizing technologies like React JS, JavaScript, Node.js, MongoDB, and various APIs, the project aims to address common issues in existing food ordering systems by offering personalized and contextually relevant user experiences. The development follows the Scrum methodology, ensuring efficient project management and timely delivery. The report also includes comparisons with similar projects, highlighting "FoodAaja's" unique features, and outlines future work plans as per the Gantt chart.

Table of Contents

1.	Introduction.....	1
1.1	Introduction to the Topic.....	1
1.2	Problem Scenario.....	2
1.3	Project as a Solution	2
1.4	Aims and Objective	3
1.4.1	Aim	3
1.4.2	Objective	3
1.5	Structure of the report	4
1.5.1	Introduction.....	4
1.5.2	Background study.....	4
1.5.3	Development to date	4
1.5.4	Analysis of progress	4
1.5.5	Future work	4
1.5.6	References	4
1.5.7	Appendix	4
2.	Background/Literature review	5
2.1	Technology used.....	6
2.1.1	UI Framework React JS	6
2.1.2	Programming Language JavaScript.....	6
2.1.3	Backend framework Node JS.....	6
2.1.4	APIs.....	6
2.1.5	Database MongoDB	6

2.2	Methodology	7
2.2.1	Considered Methodology	7
2.2.2	Selected Methodology	7
2.3	Review of similar project	9
2.3.1	Similar Project	9
2.3.2	Comparison Table Between Systems	10
3.	Development to date	11
3.1	Requirement Gathering	11
3.2	Use Case.....	12
3.3	ERD	13
3.4	SRS Document.....	14
3.5	Wireframe	14
3.6	Sequence Diagram	14
3.6.1	Sequence diagram of login process.....	14
3.6.2	Sequence diagram of Register customer process	15
3.6.3	Sequence diagram of Ordering process	16
3.6.4	Sequence diagram of payment process	17
3.6.5	Sequence diagram of Admin Adding, Editing and Removing Restaurant	18
3.6.6	Sequence diagram of Admin Adding, Editing and Removing Menu items	19
3.6.7	Sequence diagram of Admin orders view page	20
3.7	Data Flow Diagram (DFD) for system.....	21
3.7.1	Level 0 DFD	21
3.7.2	Level 1 DFD	22

3.7.3	Level 2 DFD	23
3.8	System Architecture Diagram	24
3.9	Developed features	25
3.9.1	Customer Dashboard Home Page	25
3.9.2	Restaurant Menu Page with add to card feature	25
3.9.3	Cart Page feature	26
3.9.4	Login feature	26
3.9.5	Admin Dashboard Page	27
3.9.6	Add Restaurant feature	27
3.9.7	Database Tables	28
4.	Analysis of progress	29
4.1	Overview of Current Status	29
4.2	Comparison with the Project Plan	29
4.3	Challenges Encountered	29
4.4	Successes and Achievements	29
4.5	Analysis of Development Activities	30
4.6	Action Plans	30
4.7	Reflection on Methodology	31
4.8	Progress Table	33
5.	Future Work	34
5.1	Sprint 1	34
5.2	Sprint 2	34
5.3	Sprint 3	35

5.4	Sprint 4	36
5.5	Documentation	36
5.6	System Testing.....	37
5.7	Review Project and Documentation	37
5.8	Submit the Project.....	37
6.	References	38
7.	Appendix	41
7.1	SRS Document.....	41
7.1.1	Introduction.....	41
7.1.2	Overall Description.....	42
7.1.3	Functional requirement	44
7.1.4	External Interface Requirement	49
7.1.5	Non-Functional requirements.....	50
7.2	Wireframe.....	51
7.2.1	Home Page Screen	51
7.2.2	Menu Page Screen.....	52
7.2.3	Login Screen	53
7.2.4	Register Screen.....	54
7.2.5	User Account Dropdown menu	55
7.2.6	Cart page screen.....	56
7.2.7	Check out page screen	57
7.2.8	Payment page screen	58
7.2.9	Profile page screen	59

7.2.10	Orders page screen	60
7.2.11	Admin dashboard page screen	61
7.2.12	Add restaurant screen	62
7.2.13	Menu management page screen	63
7.2.14	Add Food menu screen	64
7.2.15	Ordered list page screen	65
7.3	Survey Report	66
7.4	Considered Methodology	70
7.4.1	Prototype	70
7.4.2	Kanban	71
7.5	Selected Methodology	72
7.5.1	Scrum methodology	72
7.6	Similar Project Description	74
7.7	Progress Table	77
7.8	Technology Used	80
7.8.1	UI Framework React JS	80
7.8.2	Programming Language JavaScript	80
7.8.3	Backend framework Node JS	80
7.8.4	APIs	81
7.8.5	Database MongoDB	81
7.9	Frontend File and Folder Structure	82
7.10	Backend File and Folder Structure	82
7.11	Database connection	83

7.12 Admin Folder and Models	83
------------------------------------	----

Table of Figures

Figure 1:Market size of the global food service.....	1
Figure 2: Use case	12
Figure 3: Initial ERD of FoodAaja	13
Figure 4:Sequence diagram of login process	14
Figure 5:Sequence diagram of register customer process	15
Figure 6:Sequence diagram of ordering process.....	16
Figure 7:Sequence diagram of Payment process.....	17
Figure 8:Sequence diagram of adding, editing, and removing Restaurant	18
Figure 9: Sequence diagram of adding, editing, and removing Menu items	19
Figure 10: Sequence diagram of admin orders view page	20
Figure 11: Level 0 DFD	21
Figure 12: Level 1 DFD	22
Figure 13:Place order Level 2	23
Figure 14: Payment Process level 2	23
Figure 15: System Architecture Diagram	24
Figure 16: Customer Dashboard page.....	25
Figure 17: Menu page	25
Figure 18: Cart Page.....	26
Figure 19: Login Page.....	26
Figure 20: Admin dashboard page.....	27
Figure 21: Add restaurant form	27
Figure 22: Stored data in Database	28
Figure 23: Updated Gantt Chart.....	32
Figure 24: Updated Gantt Chart.....	32
Figure 25: System Perspective of Web Application	42
Figure 26: Wireframe of home page	51
Figure 27: Wireframe of menu page of particular restaurant	52

Figure 28: Wireframe of Login Page	53
Figure 29: Wireframe of register page	54
Figure 30: User Account Dropdown menu.....	55
Figure 31: Wireframe of cart page	56
Figure 32: Wireframe of checkout page.....	57
Figure 33: Wireframe of Payment page	58
Figure 34: Wireframe of profile page	59
Figure 35: Wireframe of orders page	60
Figure 36: Wireframe of the admin dashboard	61
Figure 37: Wireframe of add restaurant.....	62
Figure 38: Wireframe of food management page	63
Figure 39: Wireframe of add food menu.	64
Figure 40: wireframe of admin orders listing page.....	65
Figure 41: Prototype Methodology (geeksforgeeks, 2023).....	70
Figure 42: Kanban Methodology (ionos, 2023).....	71
Figure 43: Scrum Methodology (wrike.com, 2023)	72
Figure 44: Foodmandu web application user dashboard.....	74
Figure 45: Bhojdeals web application user dashboard	75
Figure 46: Bhok lagyo web application user dashboard	76
Figure 47: frontend file and folder structure	82
Figure 48: backend file and folder structure	82
Figure 49: Database connection	83
Figure 50: Frontend Admin folder	83
Figure 51: backend models	84

Table of Tables

Table 1: Comparison table between system.....	10
Table 2:Functional requirement description of customer register	44
Table 3:Functional requirement description of login	44
Table 4:Functional requirement description of food ordering	45
Table 5: Functional requirement description of Profile.....	46
Table 6:Functional requirement description of orders.....	47
Table 7:Functional requirement description of recommendation service	47
Table 8: Functional requirement description of Admin Panel.....	48
Table 9:Non-functional requirement description of performance requirements	50
Table 10:Non-functional requirement description of safety requirements	50
Table 11: Progress Table	79

1. Introduction

1.1 Introduction to the Topic

Topic: FoodAaja(food ordering web app with real-time recommendations system)

Web applications have become an essential component of business in today's world. By using web applications, businesses can now develop and become simpler, and achieve their objectives much faster (yourstory, 2022). Food ordering systems have become a staple in the tech industry, with their popularity and market size continuing to grow. I am developing a web application for food ordering services which allows the customer to easily find food items according to their preferences.

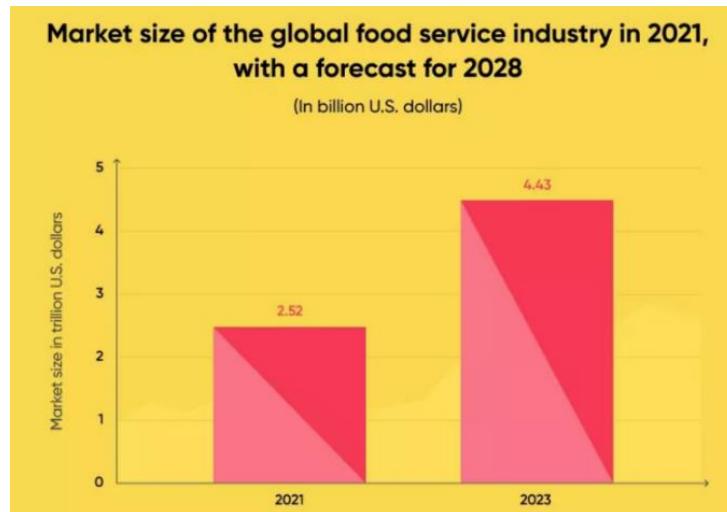


Figure 1: Market size of the global food service.

1.2 Problem Scenario

- 1) Many users are on a tight budget, but the app isn't making it easy for them to find affordable options. It doesn't offer clear suggestions for budget-friendly items or highlight choices that align with their financial constraints. Users feel like the app could be doing more to help them make smart, cost-effective decisions.
- 2) One of the common problems with food ordering apps is the lack of contextual information, such as the current weather conditions or the time of day, which can enhance the experience for instance, on a cold day, the app could suggest warm soups or comfort food to customers. Customers face problems in choosing the food items.
- 3) Customers frequently encounter difficulties when attempting to customize their orders on the food delivery app. Limited customization options and a complex user interface make users unhappy.

1.3 Project as a Solution

1. Consider a user's budget and suggest menu items or restaurants that offer affordable options.
2. By Using location data, time of day, weather conditions, and user feedback to provide real-time circumstantial recommendations. For example, the web app can suggest warm soups on a cold day. This solves the lack of contextual information.
3. Users can browse through restaurant menus, select items, and customize their orders according to their preferences.

1.4 Aims and Objective

1.4.1 Aim

The aim of this project is to develop a web application that addresses common issues faced by customers when using food ordering apps. the app will provide affordable options based on user budgets and offer contextual recommendations using real-time data like weather data. the app will allow users to browse through restaurant menus, select items, and customize their orders according to their preferences.

1.4.2 Objective

The objectives of this project are:

- 1) To understand the working mechanism of web applications.
- 2) To understand and implement a database management system in the real world with the help of a relational database.
- 3) To learn about full-stack web development.
- 4) To learn about API programming (Node.js).
- 5) Research and implementation of current tools and technology that can be integrated in web applications.
- 6) To Develop a system that suggests menu items and restaurants according to user budget.
- 7) To document the progress of the web app project and identify any challenges or risks.
- 8) To implement and learn Scrum methodology for effective project management.
- 9) To Gain a better understanding of project management principles and practices by applying them throughout this web development project.

1.5 Structure of the report

1.5.1 Introduction

An introduction to the project where the project is explained in brief. Problems that exist and ways this project helps to reduce the problems is also explained. Finally, the aims and objectives of this project are stated.

1.5.2 Background study

Background/Literature review consists of information about ends users, understanding solution for problem, review of similar application, technology and methodology that has been used in this project.

1.5.3 Development to date

In Development to date chapter, progress of project is included.

1.5.4 Analysis of progress

Analysis of progress chapter include the progress analysis to obtain information on the state of project, how it is being developed? This chapter helps to determine the actual progress of project according to Gantt chart.

1.5.5 Future work

Future work includes the remaining task of this project to be carried out in future.

1.5.6 References

In references section, it included all the references that are taken as reference while doing this project, report.

1.5.7 Appendix

It contains the brief description of SRS document, wireframes with screenshot and survey report.

2. Background/Literature review

The business world of today has been greatly impacted by the development of web applications. These applications were simple in the beginning, with only of static web pages. Now web applications have become more interactive and dynamic. (HeeSoo Chin, 2022).

Food ordering systems have grown rapidly over the last ten years, moving from traditional phone-based orders to advanced online platforms. This change started with the growth of the internet and e-commerce. which make easy to browse menu and place food order online. Originally, they were just simple online menus and phone-based orders, but now they become advanced, offering features like tracking orders in real-time, giving personalized suggestions, and easy payment options. A 2023 Netsolutions report says that the market for these services was worth 2.52 billion U.S. dollars in 2021 and is expected to reach 4.43 billion by 2028. It increase because these systems make ordering food easier and faster, with simple access to a variety of food options (fortunebusinessinsights, 2023).

2.1 Technology used

2.1.1 UI Framework React JS

In this project, React JS is used as a UI framework for the applications.

2.1.2 Programming Language JavaScript

In this project, I have chosen JavaScript as our programming language.

2.1.3 Backend framework Node JS

In this project, I have implemented Node.js as the framework for backend development.

2.1.4 APIs

a) Weather API

In this project, I have plan to use weather API to suggest food items according to the weather.

b) Map API

In this project, I have implemented map API for finding current customer locations.

2.1.5 Database MongoDB

In this project, I have chosen MongoDB as a database.

A details description of the technology used is in the appendix section i.e. [technology used](#)

2.2 Methodology

2.2.1 Considered Methodology

- a) Prototype
- b) Spiral Model

Note: The description of the considered methodology is kept in the appendix section i.e. [Considered Methodology](#)

2.2.2 Selected Methodology

a) Scrum Methodology

Scrum is an agile framework that is particularly effective for managing and completing complex projects. It emphasizes collaboration, flexibility, and iterative progress (S, 2023). This methodology has been selected for this project because of the following reasons which are listed below:

- i. This project is an academic, year-long project, there is a strict time limit, and the project must meet all user requirements within this timeframe. If the defined requirement were not fulfilled, within each time, then it would be pointless. Due to this reason, I have selected the scrum methodology for this project. Scrum uses a step-by-step method, where work is done in a set period called sprints, usually 2-4 weeks. This helps make sure that the project moves forward consistently, meets deadlines and fulfills needs (simplilearn, 2023).
- ii. In this project, the author meets with the supervisor weekly to show their project progress, which includes project reports or system development updates. During these meetings supervisors give feedback. This is similar to scrum, a methodology where teams have meetings like Sprint Review, to present their work and get feedback (scrum.org, 2023). This will help me to make quick changes based on supervisors' advice.
- iii. While developing this system, some components may share the same code. Writing the same code twice for each component could consume significant time and resources, which are particularly precious. Scrum's iterative approach helps save time and resources by allowing us to reuse code for different components (simplilearn, 2023).

- iv. Testing is important for this project to control changes in developing system. In Scrum, testing is integrated into each sprint. This continuous testing helps in the early detection and fixing of problems early, it maintains the quality of the software and manages changes well during the whole development process (Krupa, 2023).
- v. In this academic project, documentation also plays an important role. In Scrum, the main focus is on making software that works well with documentation. Documentation in Scrum is created iteratively (step-by-step), including essential details like user stories, how the system is designed, and test cases. The documents are updated regularly with each sprint. This ensures that all important details are kept up-to-date as the project moves forward (Averbuch, 2022).
- vi. In Scrum, I can keep track of all my tasks in a backlog, which helps me see what I still need to do. Scrum offers the freedom to choose the most suitable implementation for my project, it helps me to deal with changes. This method lets me work on the project step by step, check on my progress regularly, and continuous improvement (Keita, 2023).

Note: Furthermore, a detailed description of Scrum methodology is kept in appendix section i.e. [Selected Methodology](#).

2.3 Review of similar project

2.3.1 Similar Project

- a) FoodMandu
- b) Bhoj
- c) Bhok Lagyo

A Full details description of the similar project is in the appendix section i.e. [Similar Project](#)

2.3.2 Comparison Table Between Systems

S. N	Feature	FoodAaja [My Project]	Foodmandu	Bhoklagyo	Bhoj
1.	Login	Available	Available	Available	Available
2.	Register	Available	Available	Available	Available
3.	Real-time Recommendation	Available (based on weather)	Not Available	Not Available	Not Available
4.	Budget-friendly filtering option	Available	Not Available	Not Available	Not Available
5.	Order Customization	Available	Not Available	Available	Not Available
6.	User Profile Management	Available	Available	Not Available	Available
7.	Variety of cuisine	Available	Available	Available	Available
8.	Online Payment	Available	Available	Available	Available
9.	Offline Payment	Available	Available	Available	Available
10.	Location selection using map	Available	Available	Not Available	Available
11.	Meal Type Search filters	Available	Not Available	Not Available	Available

Table 1: Comparison table between system

The comparison table shows that my project, "FoodAaja," will include unique features not present in existing systems. These include real-time food recommendations based on current weather, a budget-friendly filter for economical meal choices, and an order customization option, which is currently only available in Bhoklagyo but missing from Foodmandu and Bhoj.

3. Development to date

3.1 Requirement Gathering

The requirement of this project is gathered through online survey and internet research. the survey was created using Google Forms and conducted from December 18 to December 29, 2023.

According to the survey, 83.4% of people who order food online spend less than Rs 3000. People are interested in a feature that suggests food options within their budget Because a feature would help people find meals that match their spending limits. For instance, someone with a tight budget could find affordable options. About 58.3% of the people usually get exactly what they order, but not always. 8.3% rarely get what they ordered. This shows that while orders are often correct, they could be more accurate.

A large number, 75% of the people find it very important to be able to customize their orders, such as adding or removing ingredients. This shows a strong customer need for flexible customization options in online food ordering services so, I am implementing a feature that allows customers to input their specific customization details. The survey indicates that there is often confusion among users when choosing food items. To address this, I'm adding a feature that recommends dishes based on real-time data such as the weather.

All the surveys with their graph are kept in the appendix section i.e. [online survey](#)

3.2 Use Case

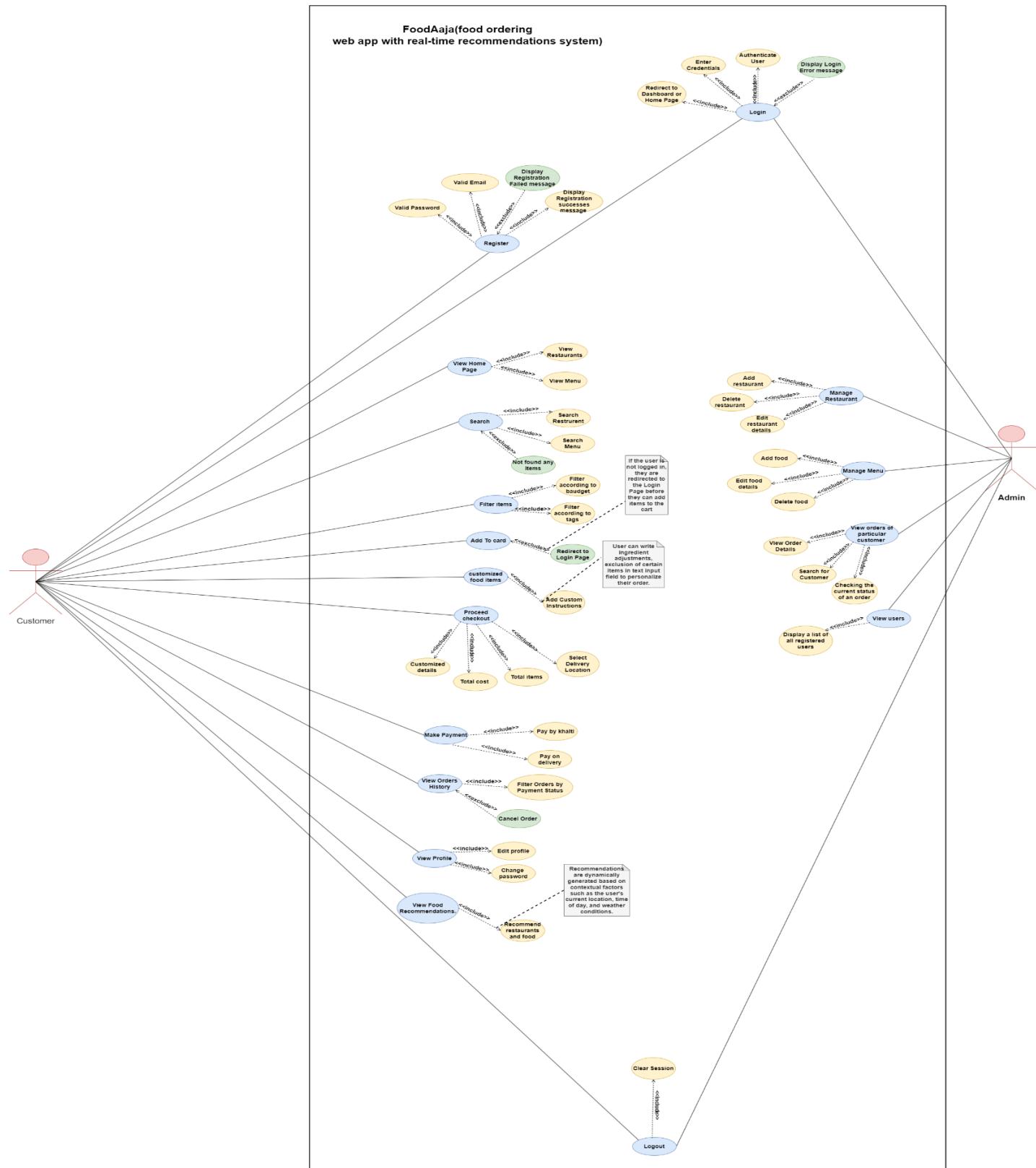


Figure 2: Use case

3.3 ERD

Initial ERD of FoodAaja

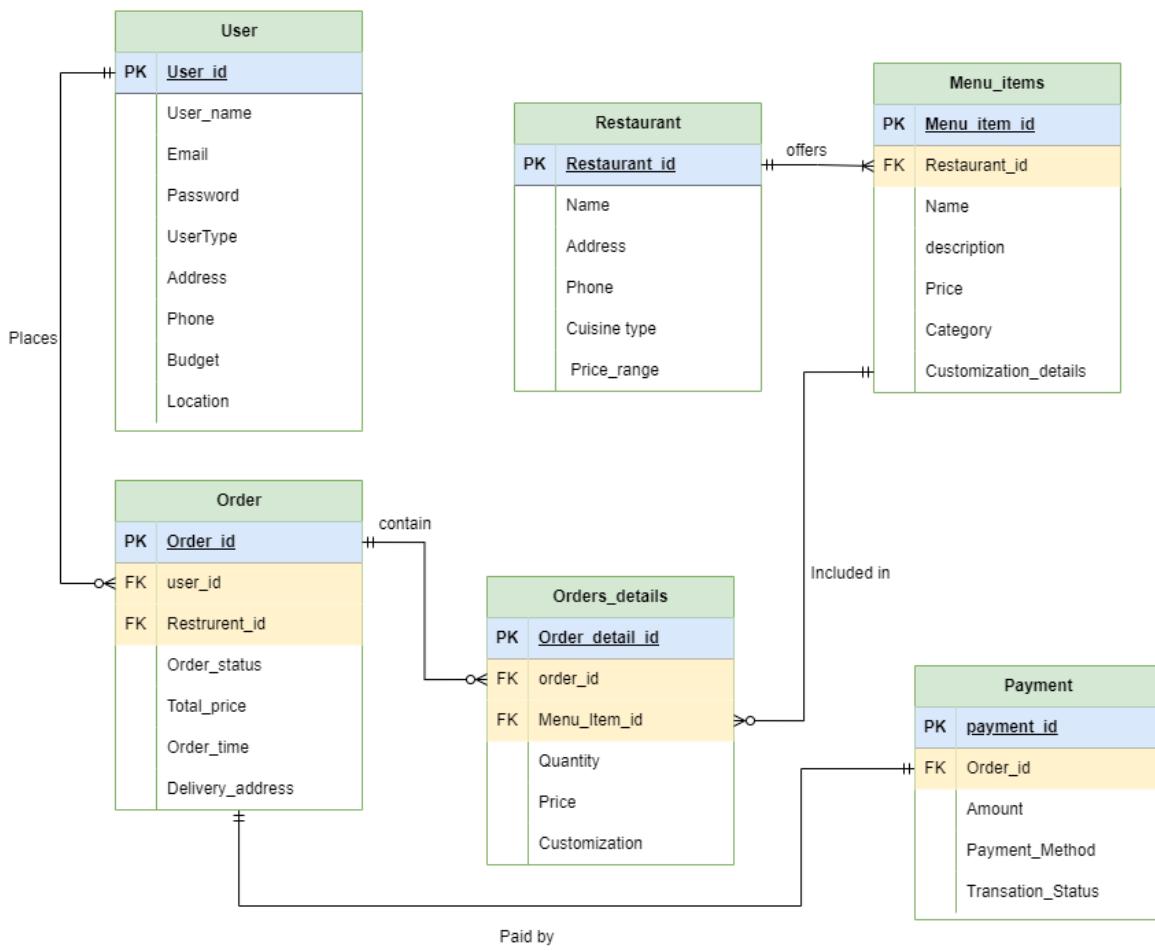


Figure 3: Initial ERD of FoodAaja

3.4 SRS Document

The full description of SRS document is kept in Appendix section i.e. [SRS Document](#).

3.5 Wireframe

The images of wireframes are kept in Appendix section i.e. [Wireframes with its full description](#).

3.6 Sequence Diagram

3.6.1 Sequence diagram of login process

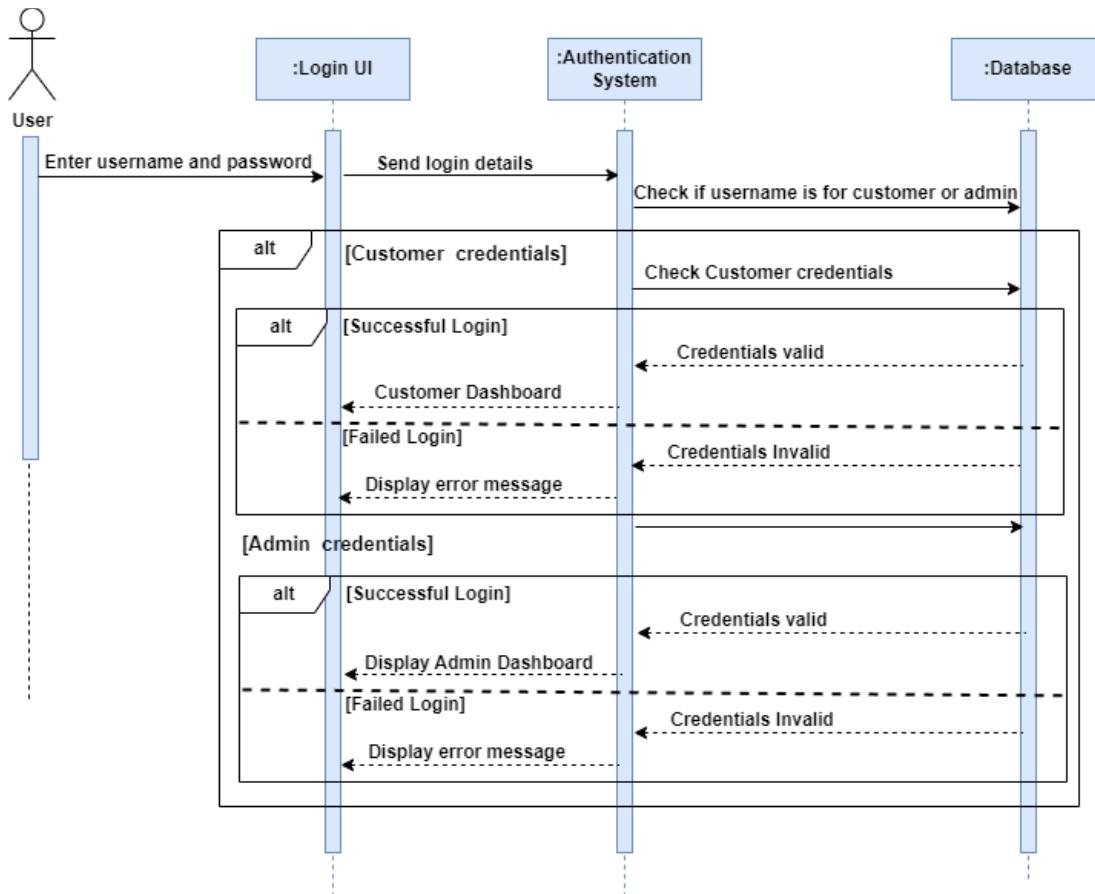


Figure 4: Sequence diagram of login process

3.6.2 Sequence diagram of Register customer process

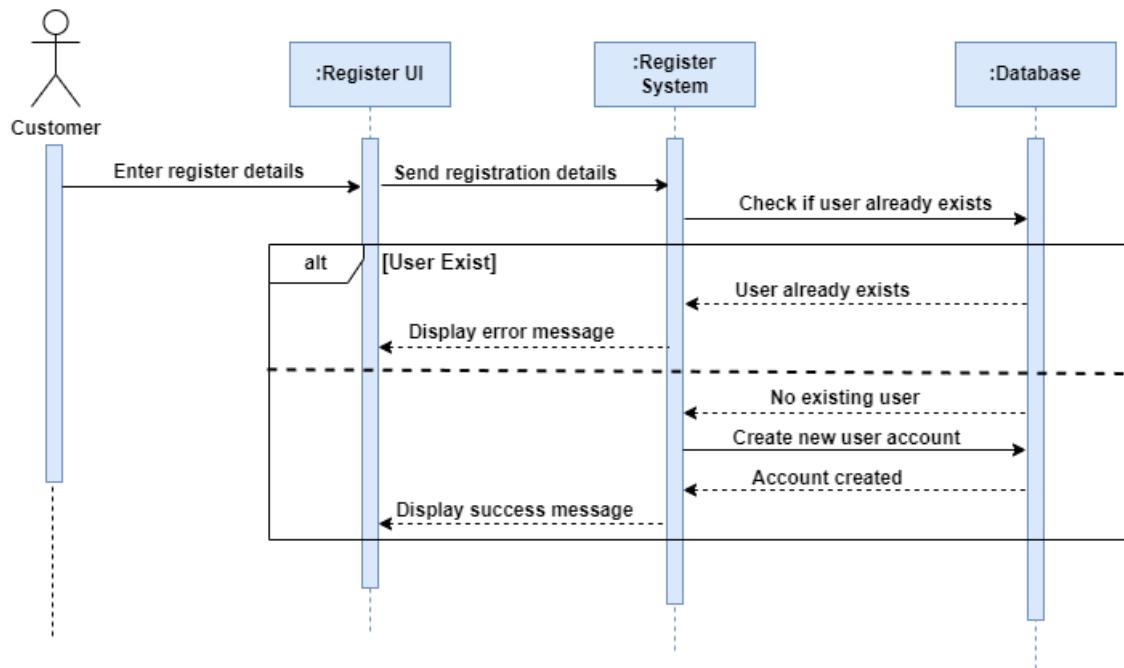


Figure 5: Sequence diagram of register customer process

3.6.3 Sequence diagram of Ordering process

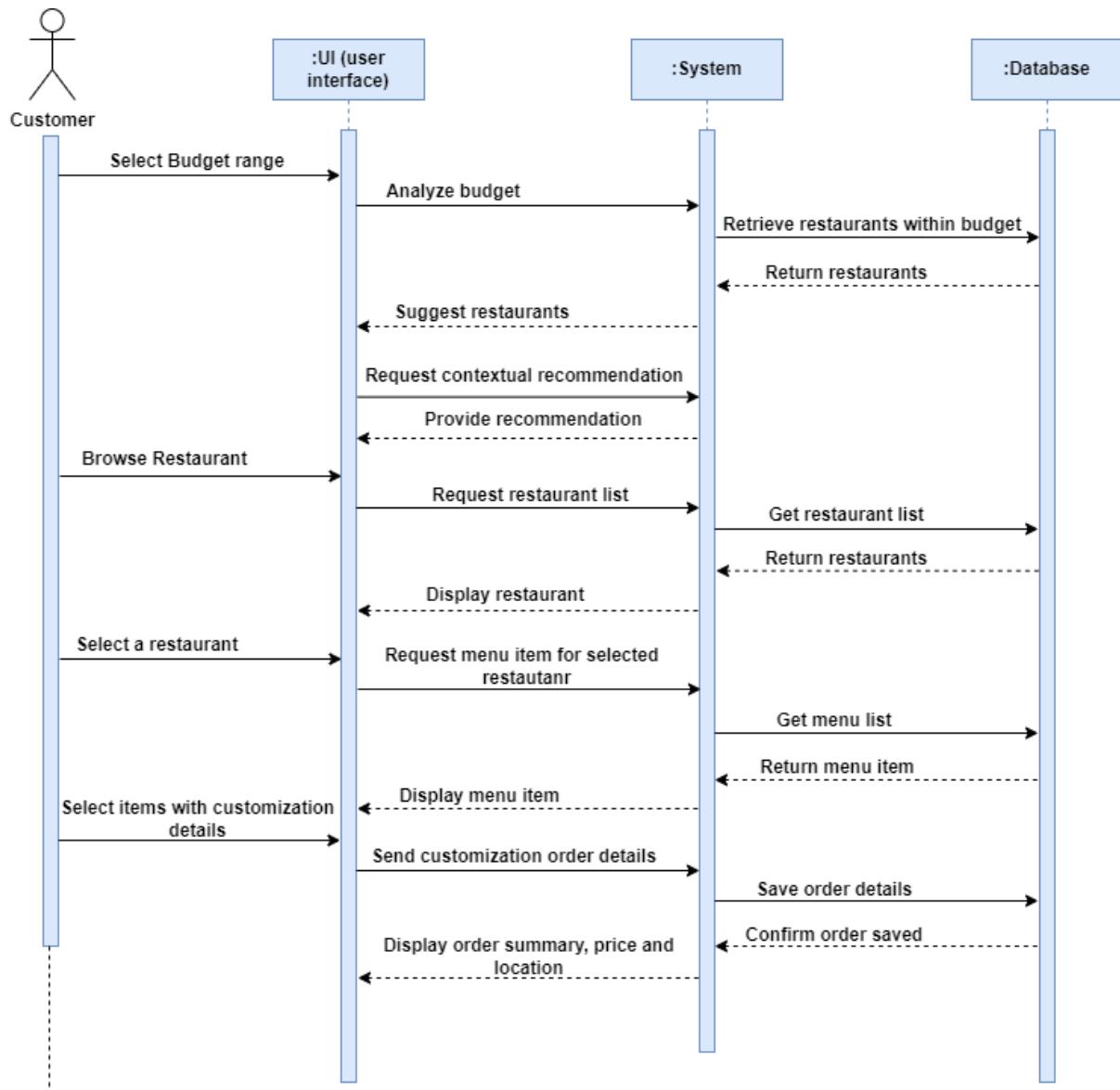


Figure 6: Sequence diagram of ordering process

3.6.4 Sequence diagram of payment process

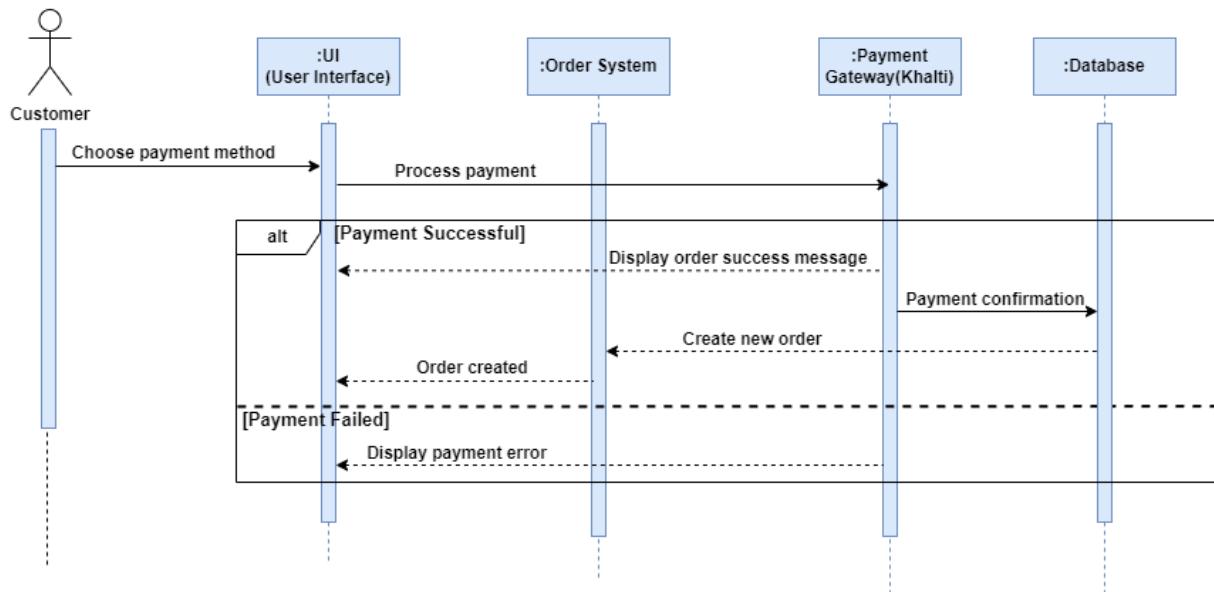


Figure 7: Sequence diagram of Payment process

3.6.5 Sequence diagram of Admin Adding, Editing and Removing Restaurant

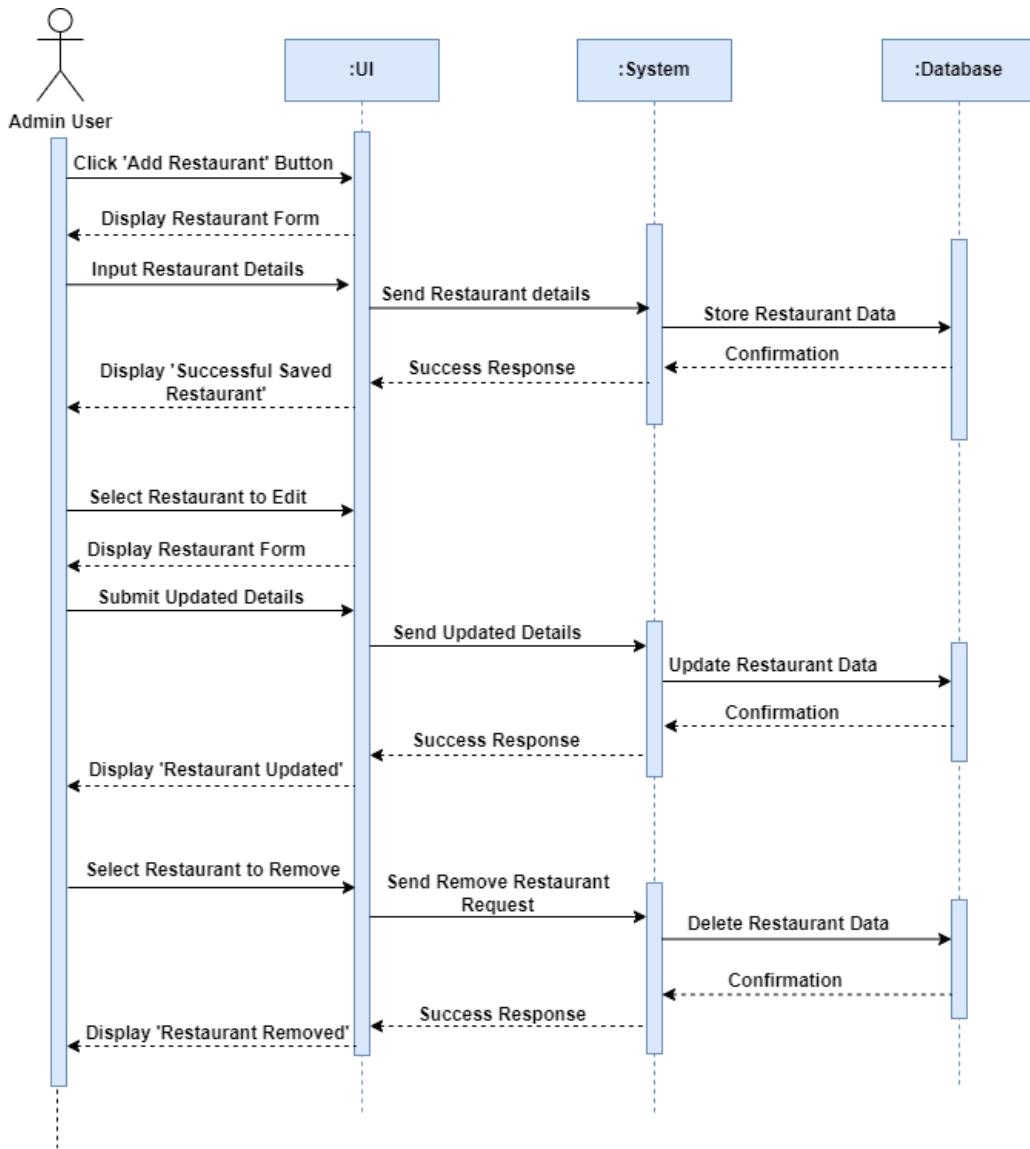


Figure 8: Sequence diagram of adding, editing, and removing Restaurant

3.6.6 Sequence diagram of Admin Adding, Editing and Removing Menu items

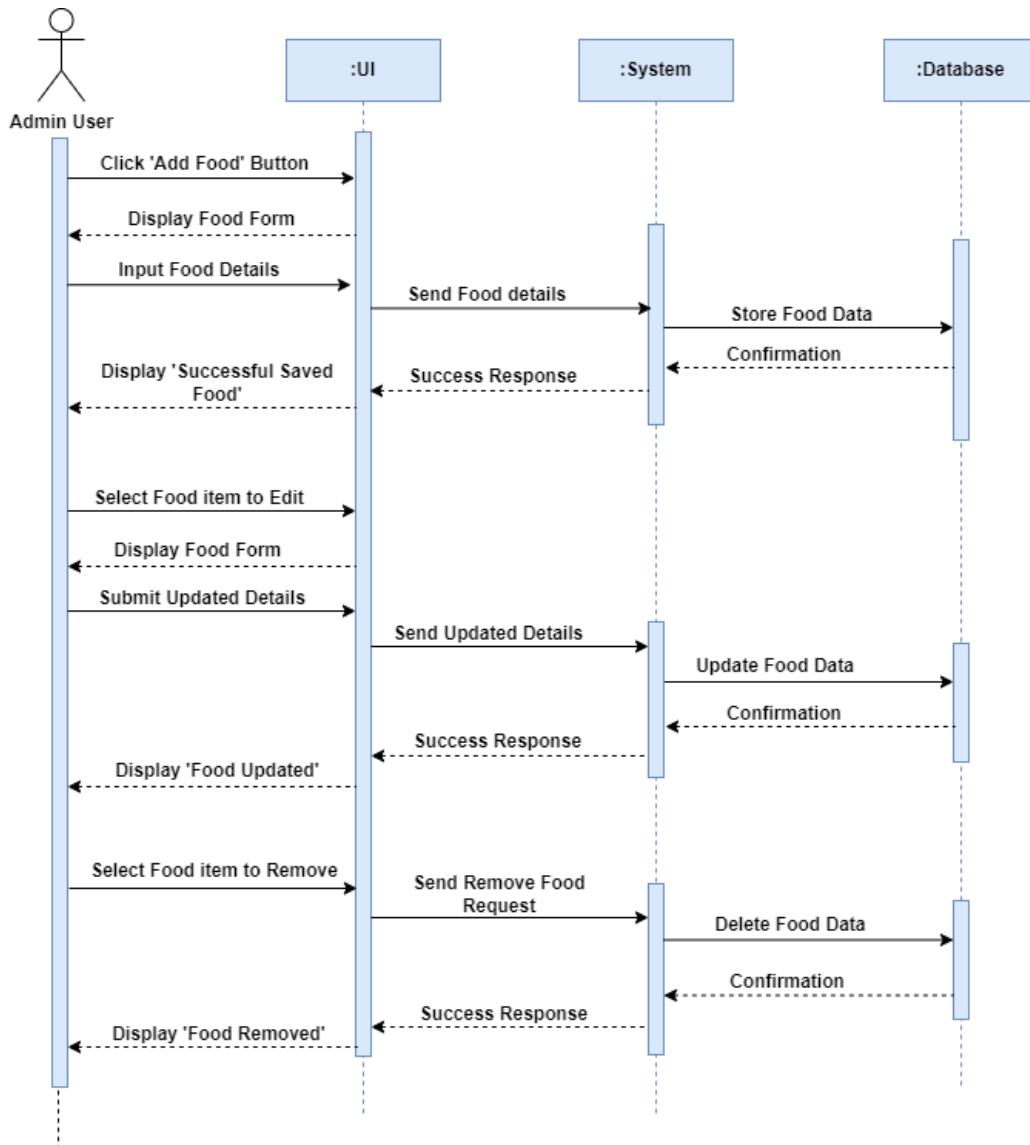


Figure 9: Sequence diagram of adding, editing, and removing Menu items

3.6.7 Sequence diagram of Admin orders view page

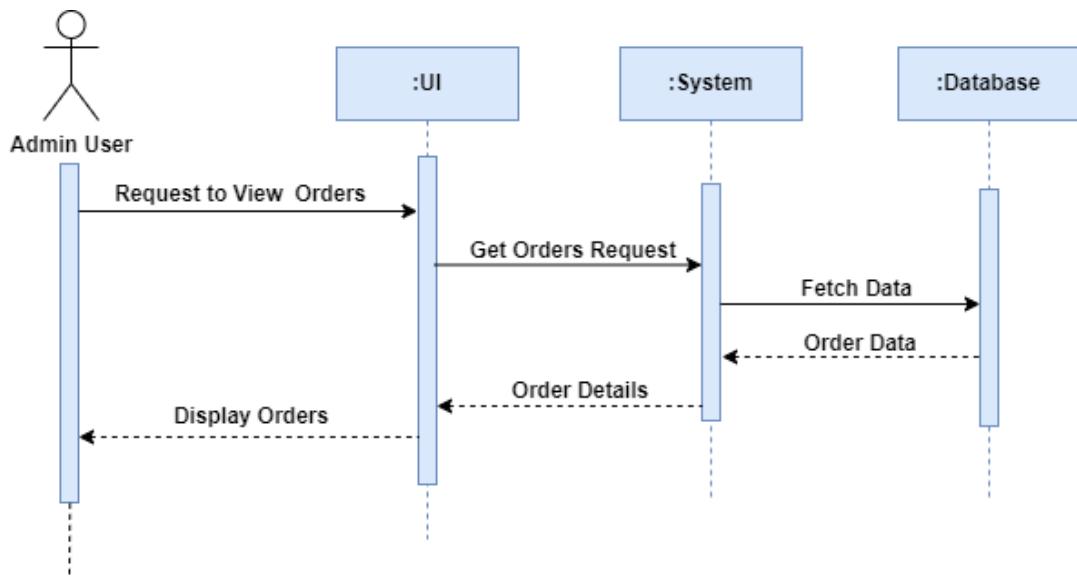


Figure 10: Sequence diagram of admin orders view page

3.7 Data Flow Diagram (DFD) for system

3.7.1 Level 0 DFD

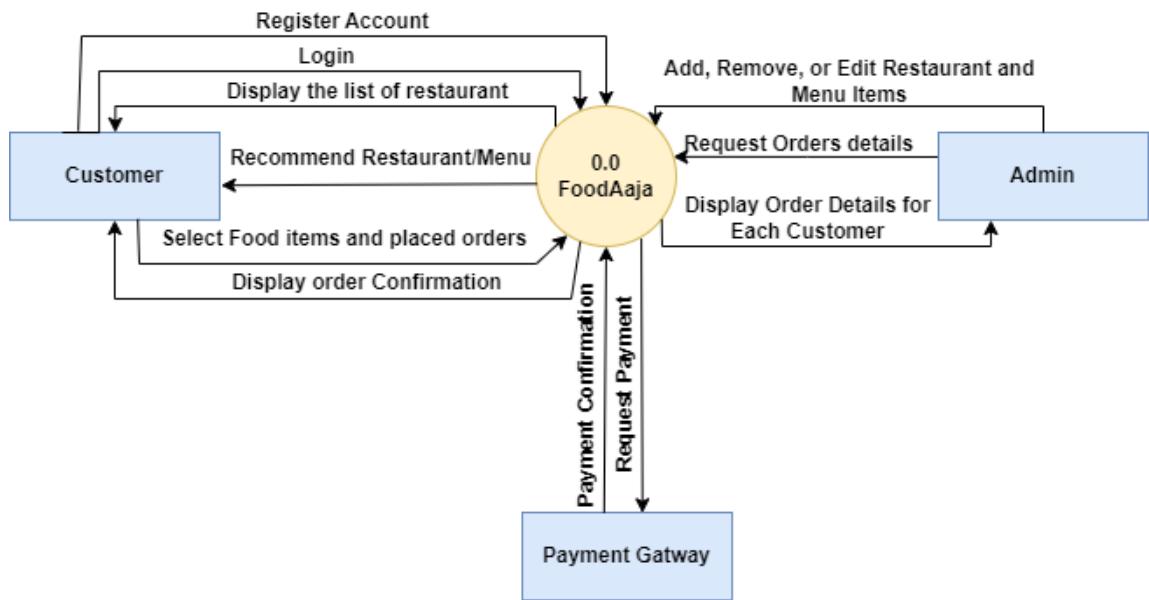


Figure 11: Level 0 DFD

3.7.2 Level 1 DFD

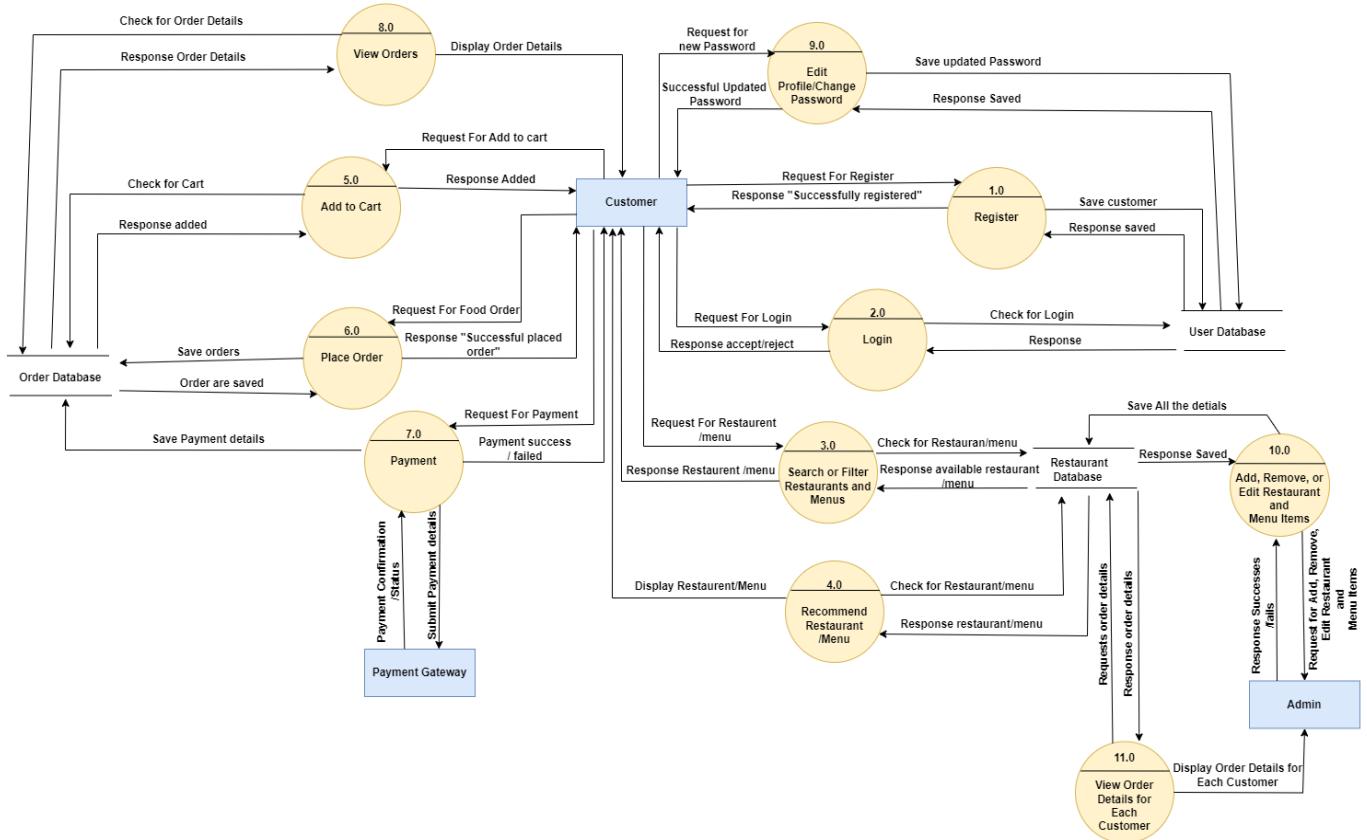


Figure 12: Level 1 DFD

3.7.3 Level 2 DFD

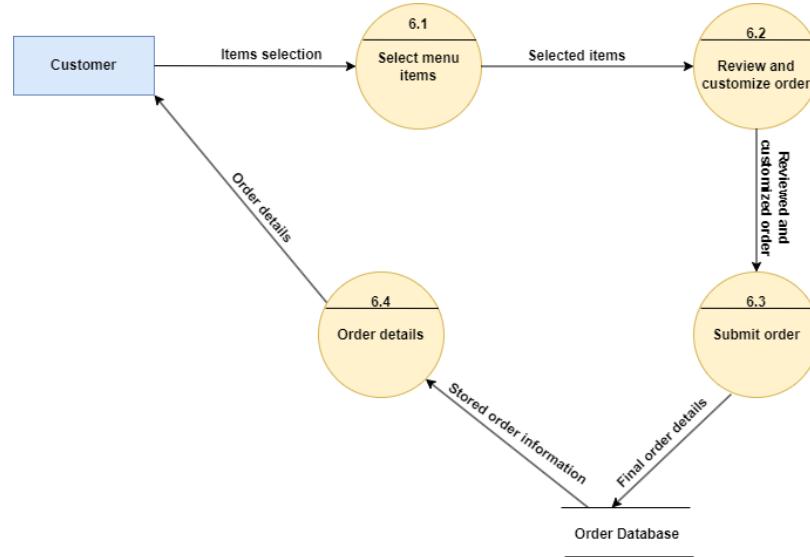


Figure 13:Place order Level 2

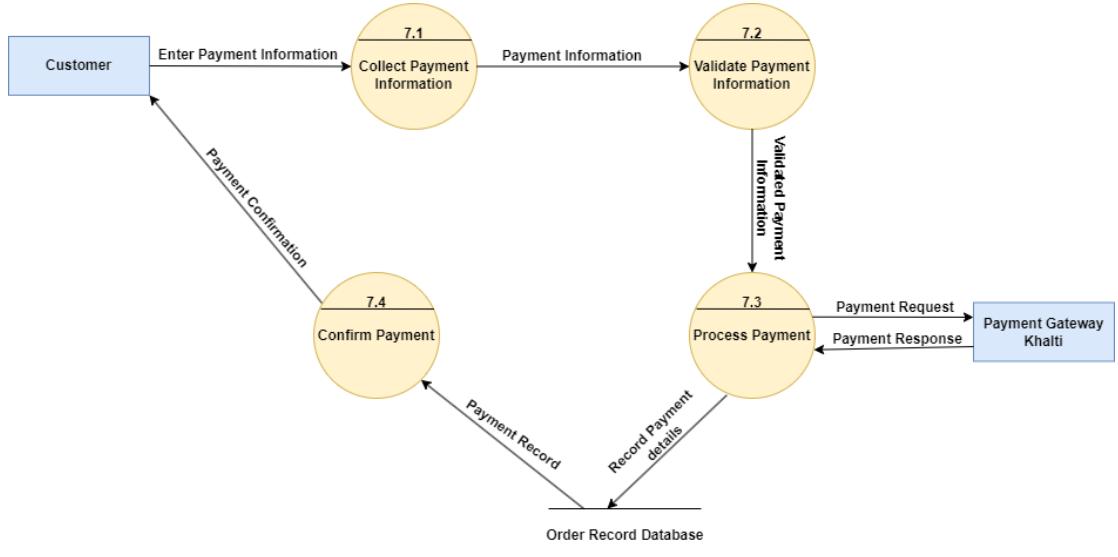


Figure 14: Payment Process level 2

3.8 System Architecture Diagram

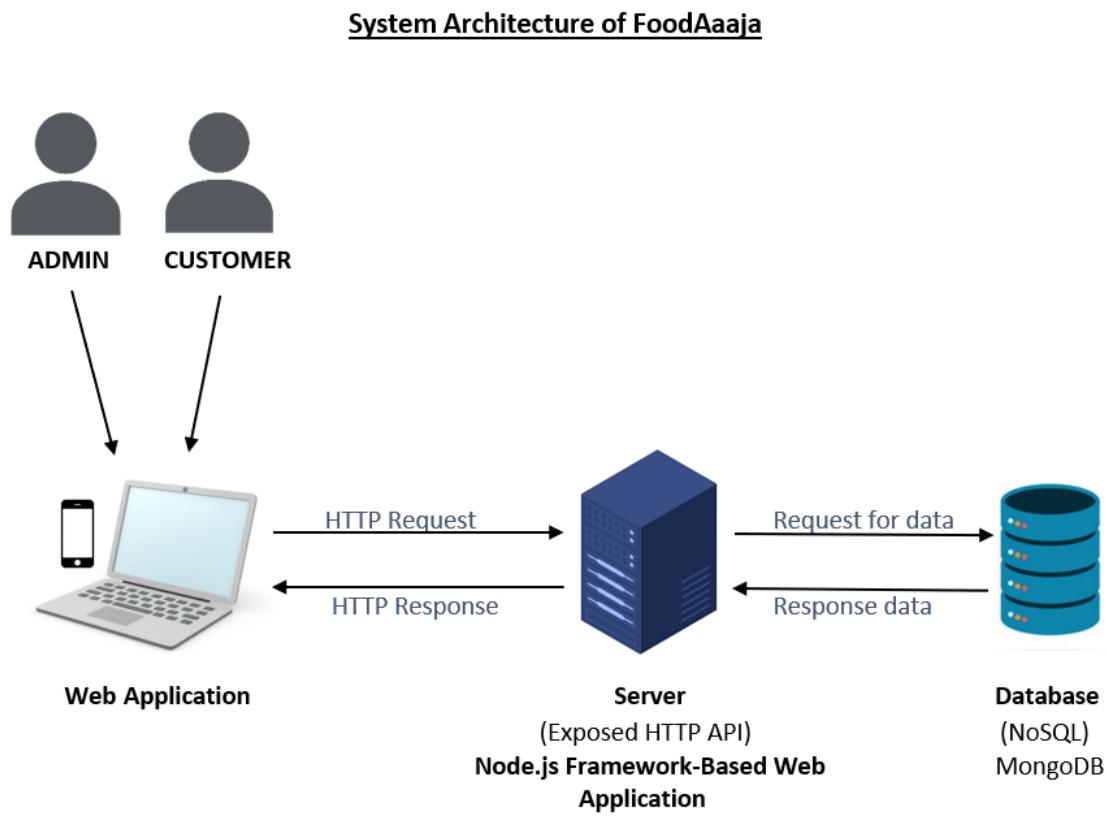


Figure 15: System Architecture Diagram

In a web application, admins and customers interact by sending HTTP requests from their devices. The server processes these requests, either fetching data from the database or responding back to the application. This process displays the relevant information to either the admin or the customer. The server plays a crucial role in facilitating communication between the user interface (UI) and the database.

3.9 Developed features

3.9.1 Customer Dashboard Home Page

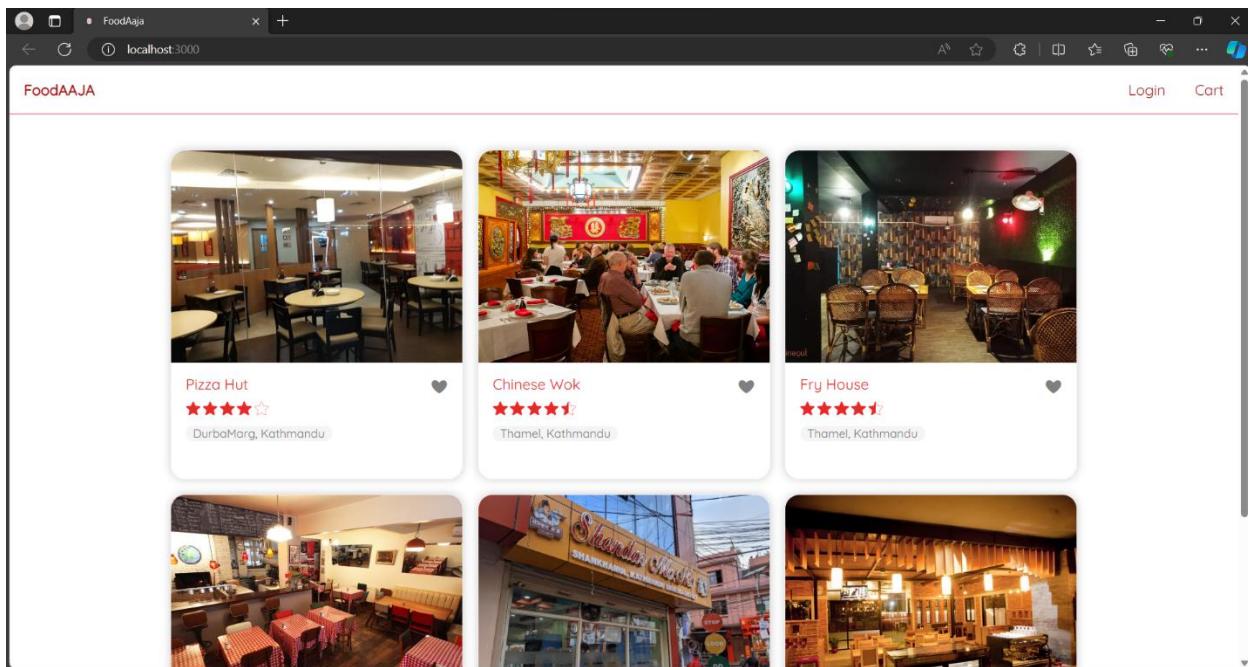


Figure 16: Customer Dashboard page

3.9.2 Restaurant Menu Page with add to card feature

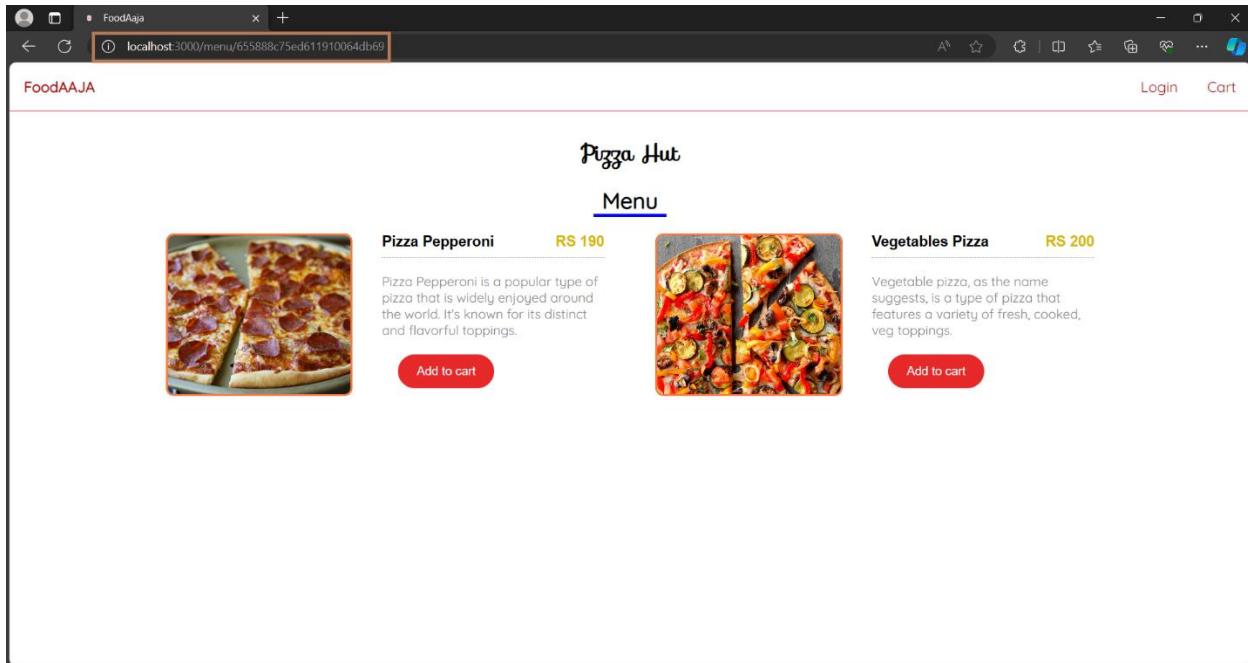


Figure 17: Menu page

3.9.3 Cart Page feature

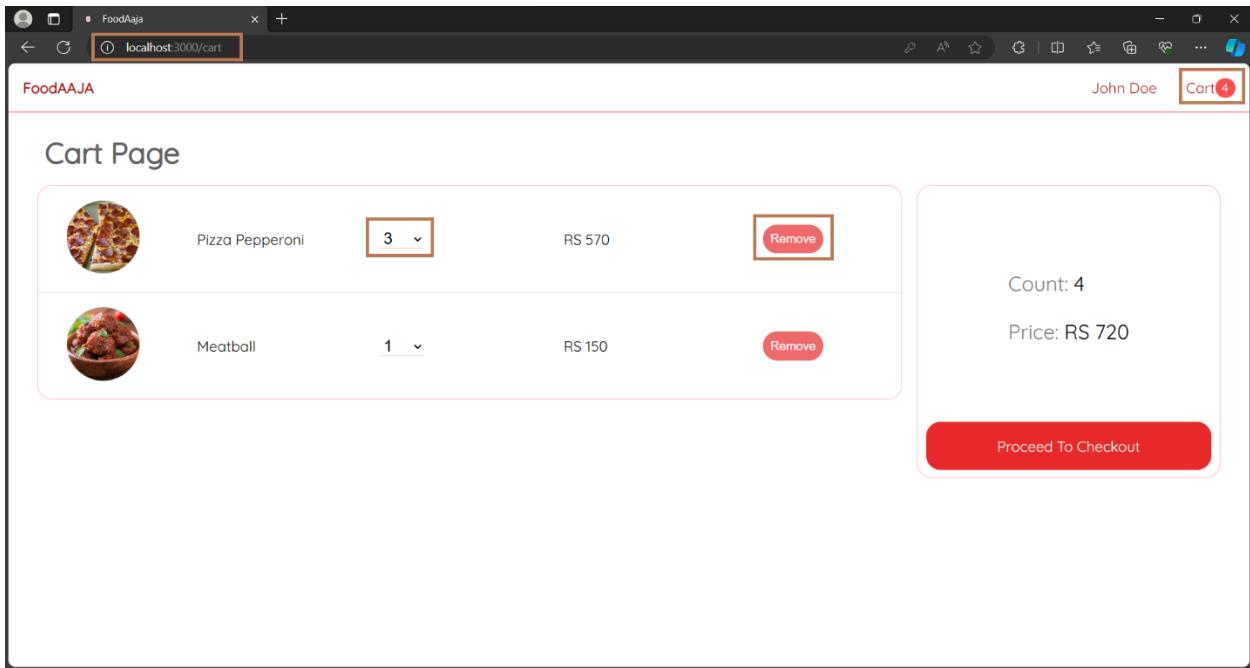


Figure 18: Cart Page

3.9.4 Login feature

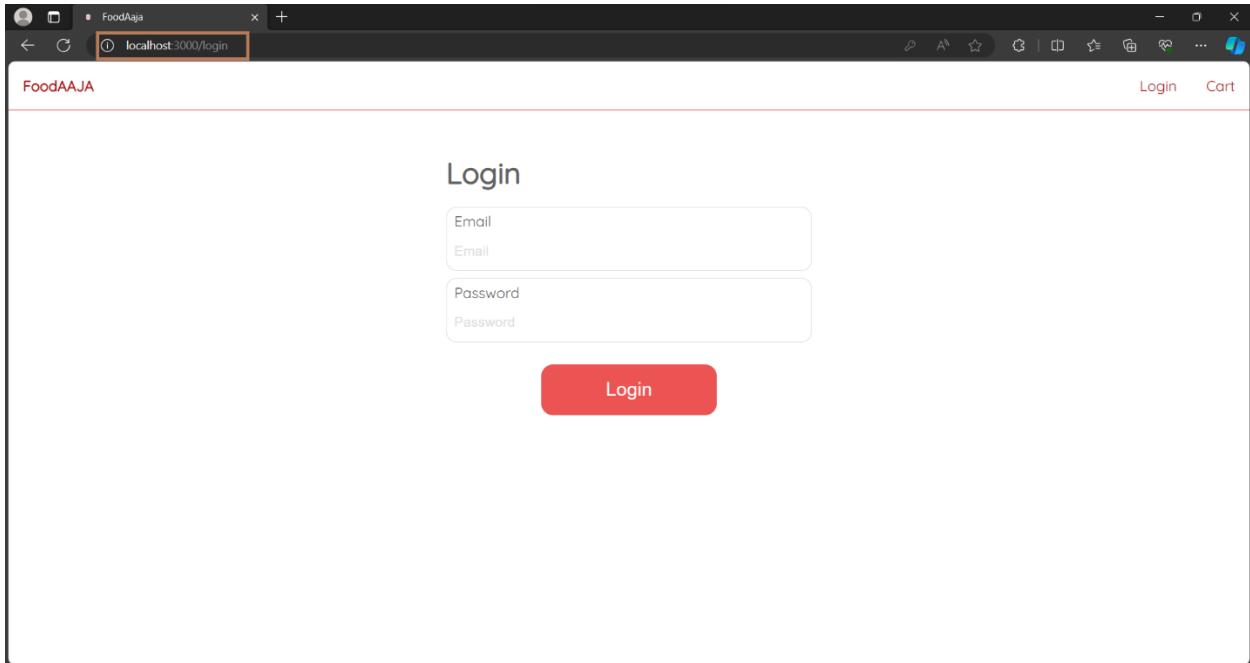


Figure 19: Login Page

3.9.5 Admin Dashboard Page

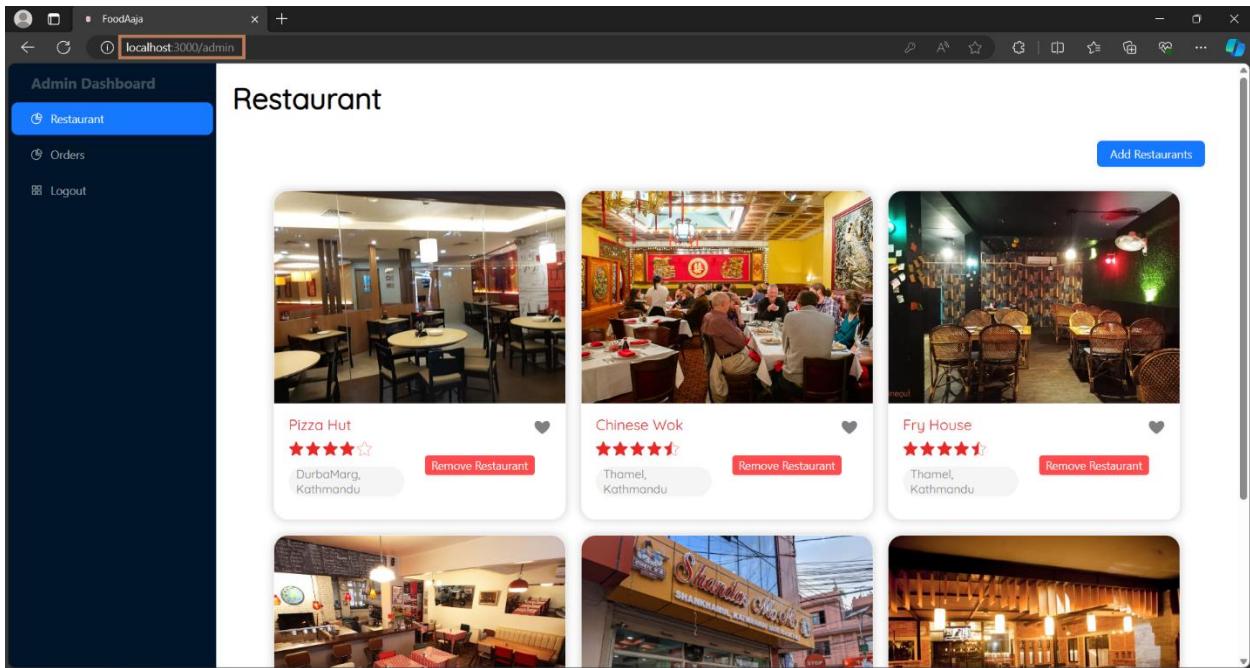


Figure 20: Admin dashboard page

3.9.6 Add Restaurant feature

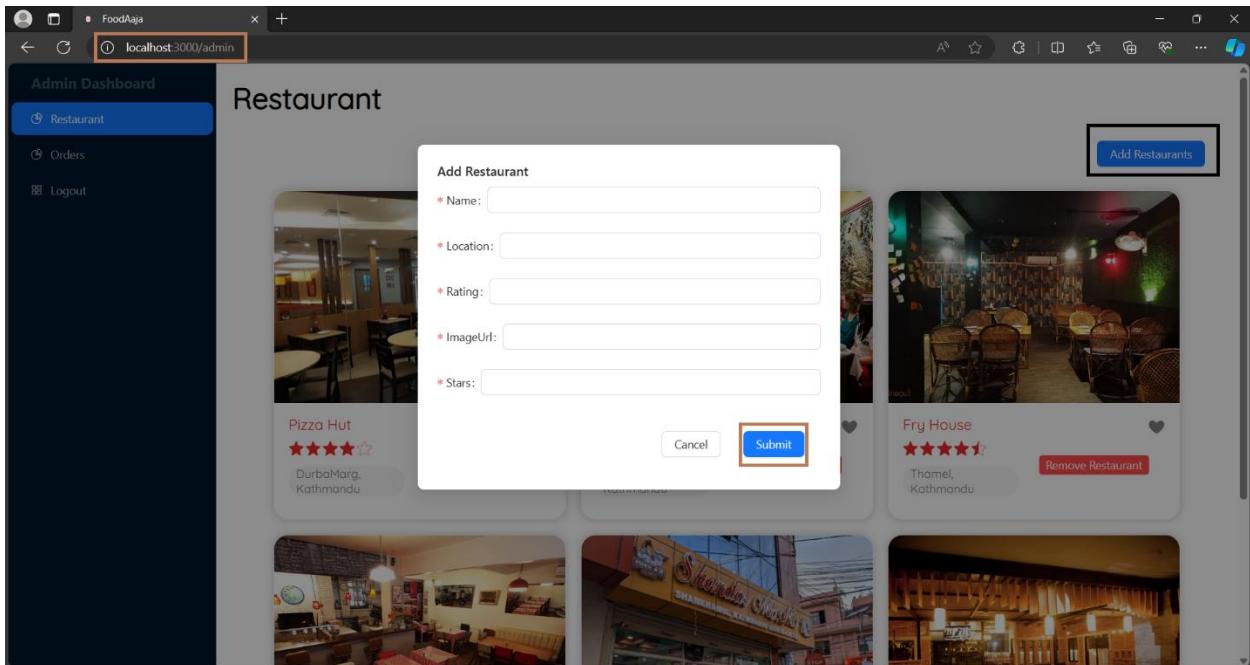


Figure 21: Add restaurant form

3.9.7 Database Tables

The screenshot shows the MongoDB Compass interface connected to the host 127.0.0.1:27017. The left sidebar lists databases: admin, config, foodaja-db2 (selected), and local. The 'Collections' section displays two collections: 'restaurants' and 'users'. The 'restaurants' collection has a storage size of 20.48 kB, 6 documents, an average document size of 612.00 B, 1 index, and a total index size of 36.86 kB. The 'users' collection has a storage size of 20.48 kB, 3 documents, an average document size of 225.00 B, 2 indexes, and a total index size of 73.73 kB.

Collection	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
restaurants	20.48 kB	6	612.00 B	1	36.86 kB
users	20.48 kB	3	225.00 B	2	73.73 kB

Figure 22: Stored data in Database

4. Analysis of progress

4.1 Overview of Current Status

As of the December 2023, the “FoodAaja” project has successfully completed several initial phases, including requirement gathering, development of Use case, development of initial ERD (Entity-Relation Diagram), successfully created the SRS (Software Requirement Specification) document. Successfully created wireframes for the Application. The requirement was also gathered from survey, which show the important needs of feature such as ability to customize orders, budget friendly option choices and food recommendation. The project is moving in progress according to the planned schedule.

4.2 Comparison with the Project Plan

Compared to the original project plan shown in the Gantt chart, the development is on track. The requirement gathering and initial design phases were completed on time. Additionally, I have half completed the restaurant and menu listing page, and 'add to cart' functionality. Furthermore, the user login feature has also been fully implemented.

4.3 Challenges Encountered

I encountered challenges while using Node.js for the backend development, as it was my first time working with this technology. Using MongoDB as the database was challenging because I was not familiar with its structure and how it works. Managing time for starting the project was challenging, as I had to balance it with other projects from different courses. I had to plan and prioritize carefully.

4.4 Successes and Achievements

One of my major achievements was quickly learning new technologies. In a short time, I taught myself React and Node.js from different sources and started the project with them. I was successful in starting this project even though, I had to handle different projects from other courses simultaneously. I have

successfully achieved several key milestones in the project, including the development of the initial wireframes and the initial UI. I have implemented login and add to card functionality. I have successfully developed the UI of the project, including the restaurant and menu listing pages, the login function, the admin dashboard, and the cart page.

4.5 Analysis of Development Activities

The activities completed so far, such as user surveys and creating the SRS documentation, have been effective in guiding the development process. The survey gave important information about what customers want, which helped shape the application features. I used the Scrum methodology for this project, and it is very effective.

4.6 Action Plans

Due to the challenges, especially with adding complex features and keeping to a tight schedule, I've taken several steps to get back on track. I am focused on completing the most important features first.

I am simultaneously developing the frontend and backend parts, which allows me to quickly implement any changes, if any problems come up, it's easier to fix them since I can work on both the frontend and backend together. I am dedicating about 13 hours each week to ensure the project, including all features and documentation, is completed before the deadline. To work more efficiently, I'm using better development tools and getting advice from experienced professionals.

Regularly reviewing the project with the supervisor, helps in quickly resolving issues and allows me to progress to the next functionality without delay.

4.7 Reflection on Methodology

In this project, using the Scrum methodology made a big difference. It helped me manage tasks more efficiently, especially during complex parts of development. By working in sprints, an important part of Scrum, I could concentrate on certain tasks within set periods, which helped me to meet deadlines. Regular Scrum meetings with my supervisor ensured that every part of the project matched our main goals, and I could quickly fix any issues. Each sprint was dedicated to specific tasks, which made the project more manageable and organized.

The detailed analysis of the sprint progress is described below:

Sprint 1

I have already started Sprint 1 and have completed several tasks, including setting up the server infrastructure and database connections. The customer login, along with authentication and authorization, is operational. Both the Restaurant Listing Page and the Menu Items Listing are halfway completed. The remaining tasks are the 'Add, Edit, Remove Restaurant' feature and the 'Add, Edit, Remove Menu Item' feature. I am still in the Sprint 1 phase of development, and so far, everything is moving forward as planned.

The updated Gantt chart is shown below:

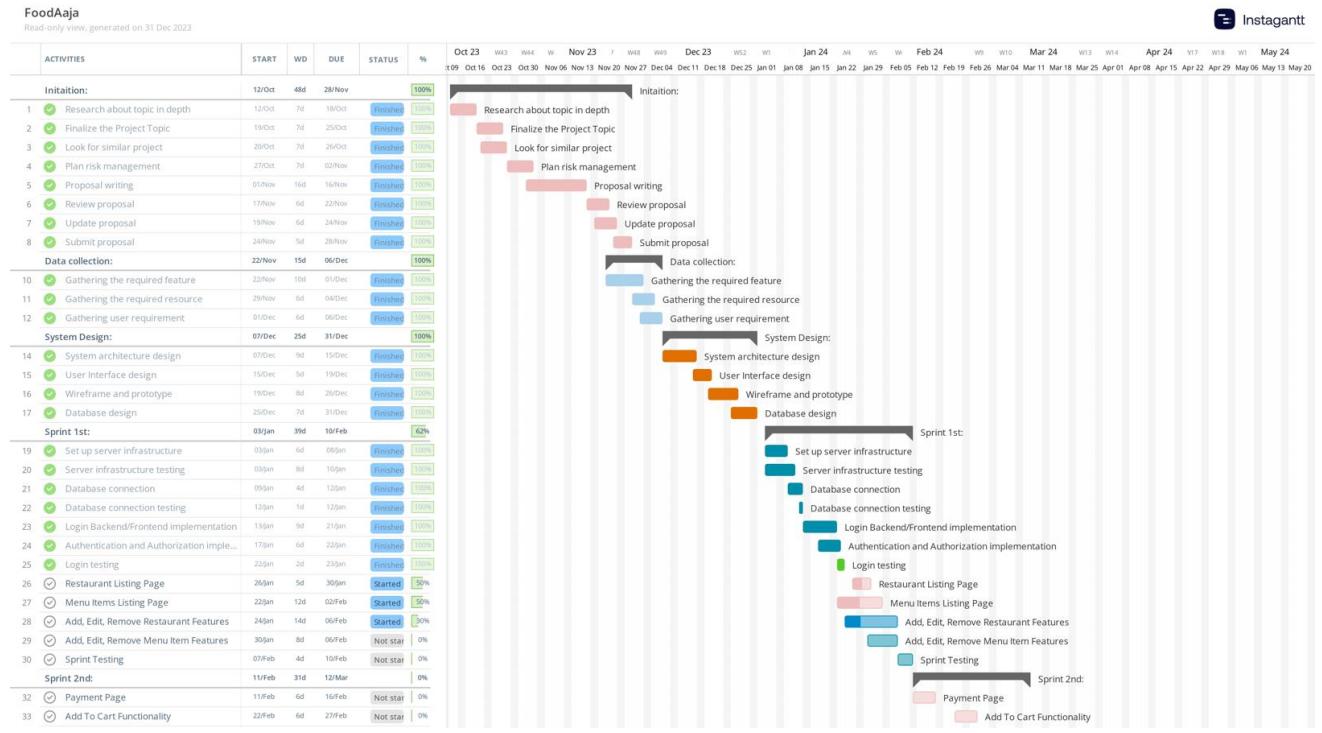


Figure 23: Updated Gantt Chart

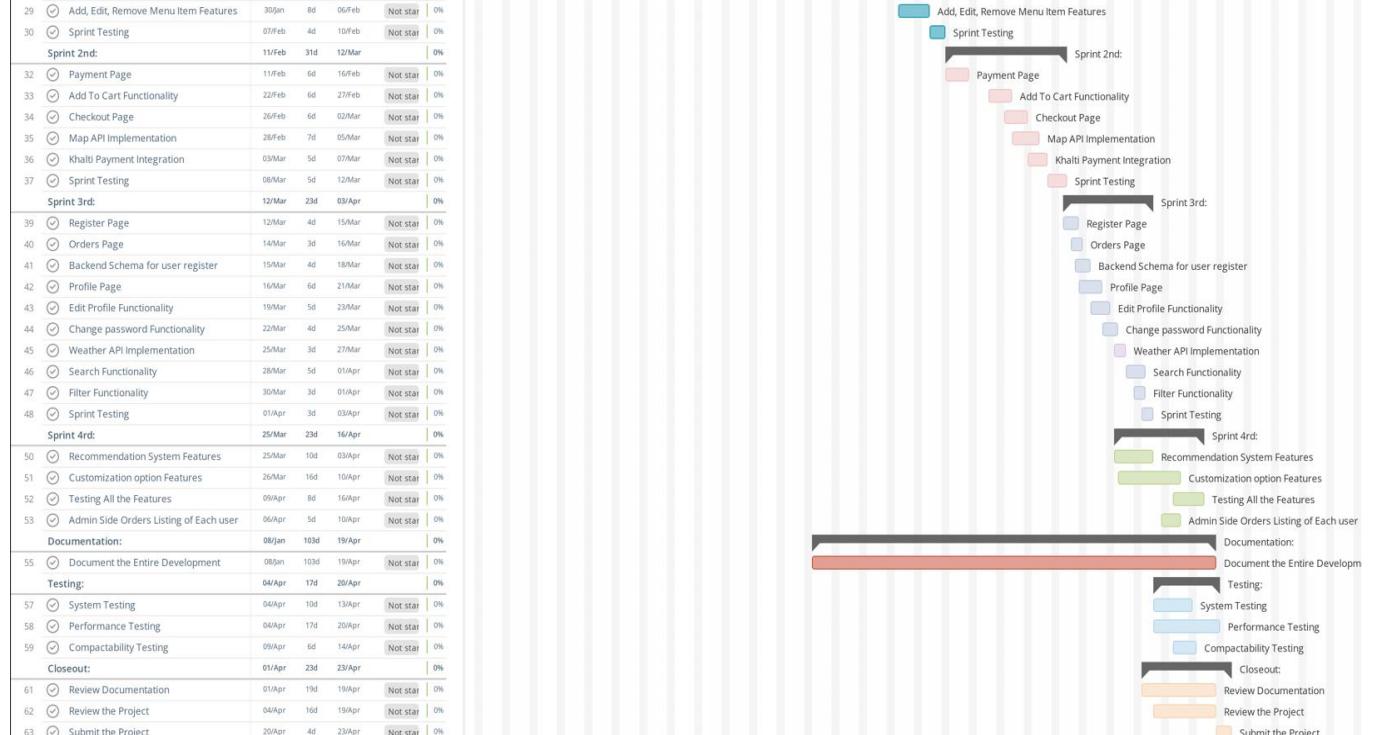


Figure 24: Updated Gantt Chart

4.8 Progress Table

Analysis of progress sections includes the progress of final year project and how it is developing? This section helps to determine the actual progress of project by comparing with Gantt Chart.

The progress Table is in appendix section i.e. [Progress Table](#)

5. Future Work

Future work includes the remaining task of project that is to be done in future within a time estimated in Gantt Chart. The sections below describe the future work for each as shown in the chart.

5.1 Sprint 1

1. Restaurant Management Features

I am currently in Sprint 1, where I have completed the 'Add Restaurant' feature. My next action will be to complete the development of features for editing restaurant details and deleting restaurants from the platform. Menu management for each restaurant has not been started yet, but I plan to begin as soon as the interim report is complete.

2. Sprint Testing

Once I finish all the features in this sprint, I will do sprint testing for the last step. testing each feature carefully to make sure everything is working well.

5.2 Sprint 2

1. Payment and Checkout Development

After completing the Restaurant Management Features, my next task will be to create a Checkout Page and a Payment Page. The goal is to ensure that users can easily pay and finalize their orders with just a few clicks once they decide what they want to eat.

2. Map API Implementation

After completing the Payment and Checkout Page development, my plan is to start integrating the Map API. This will let users easily choose where they want their food delivered.

3. Khalti Payment Integration

After I complete the Checkout Page, I plan to implement the Khalti Payment system as a payment method. This will allow our users to pay online.

4. Sprint Testing

Once I finish all the features in this sprint, I will do sprint testing for the last step. testing each feature carefully to make sure everything is working well.

5.3 Sprint 3

1. User Registration and Profile Management

After completing Sprint 2, I will start Sprint 3, focusing on User Registration and Profile Management, including development of the Register, Orders, and Profile pages, and features for editing profiles and changing passwords

2. Weather API Integration

After successfully implementing User Registration and Profile Management, I will move on to my second task for Sprint 3, which is the Weather API Integration.

3. Search and Filter Development

After successfully completing the Weather API Integration, my next step, according to the schedule in the Gantt chart, will be to start on the Search and Filter Development.

4. Sprint Testing

Once I finish all the features in this sprint, I will do sprint testing for the last step. testing each feature carefully to make sure everything is working well.

5.4 Sprint 4

1. Recommendation System Implementation

After I have completed all the tasks in Sprint 3, my next step will be to develop the Recommendation System Implementation. This will involve building and testing 'Recommendation System Features' that use weather API.

2. Customization Options

After I finish the Recommendation System Implementation, my next task will be to work on implementing 'Customization Option Features'. This will let users customize their orders based on their own taste and dietary needs.

3. Admin Order Management

After completing the Customization Option Feature, as scheduled in the Gantt chart, I will move on to the Admin Order Management task. This task will involve creating a system that allows admins to easily view each users list of orders.

4. Sprint Testing

Once I finish all the features in this sprint, I will do sprint testing for the last step. testing each feature carefully to make sure everything is working well.

5.5 Documentation

I will start the documentation phase of my project from January 8th, as planned in my Gantt chart. Documentation is important and should start early in the development process. After completing the interim report, I will begin the documentation part.

5.6 System Testing

System Testing is an important part of the project. I need to make sure that every feature of my project functions correctly to prevent any issues or crashes, especially during the viva presentation. After completing the development of the project, I will start testing the entire system.

5.7 Review Project and Documentation

After completing the development and documentation of the project, I will conduct a final review with my supervisor. This step is important to gather feedback and make necessary updates.

5.8 Submit the Project

In the given deadline, I will submit my final year project to MST. The submission will include a documentation report in PDF format and folders such as Development.

6. References

- atlassian, 2023. *Scrum Methodology*. [Online]
Available at: <https://www.atlassian.com/agile/scrum>
[Accessed 11 6 2023].
- Averbuch, Z., 2022. *scrum-documentation*. [Online]
Available at: <https://monday.com/blog/monday-workdocs/scrum-documents/>
[Accessed 11 12 2023].
- aws.amazon, 2023. *scrum methodology*. [Online]
Available at: <https://aws.amazon.com/what-is/scrum/#:~:text=Scrum%20is%20a%20management%20framework,experience%2C%20and%20adapt%20to%20change>.
[Accessed 28 12 2023].
- developer.mozilla.org, 2023. *JavaScript*. [Online]
Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
[Accessed 29 12 2023].
- ekbana, 2023. *Technology used in foodmandu*. [Online]
Available at: <https://www.ekbana.com/case-studies/foodmandu-online-food-ordering-app>
[Accessed 22 12 2023].
- fortunebusinessinsights, 2023. *food service market*. [Online]
Available at: <https://www.fortunebusinessinsights.com/food-service-market-106277>
[Accessed 25 12 2023].
- geeksforgeeks, 2023. *software-prototyping-model*. [Online]
Available at: <https://www.geeksforgeeks.org/software-prototyping-model-and-phases/>
[Accessed 11 12 2023].
- Ghimire, A., 2023. *Bhok Lagyo*. [Online]
Available at: <https://english.onlinenhabar.com>this-startup-will-hear-your-call-when-you-get-hungry-at-midnight.html>
[Accessed 22 12 2023].
- Gillis, A. S., 2023. *about MongoDB database*. [Online]
Available at: <https://www.techtarget.com/searchdatamanagement/definition/MongoDB>
[Accessed 29 12 2023].
- HeeSoo Chin, D. P. M. & D. L., 2022. Digital transformation trends in service industries. *springe*, 11(3), p. 26.
- infoq.com, 2023. *Scrum methodology for individual*. [Online]
Available at: <https://www.infoq.com/news/2015/02/personal-scrum/>
[Accessed 10 6 2023].
- ionos, 2023. *about-kanban*. [Online]
Available at: <https://www.ionos.ca/digitalguide/websites/web-development/about-kanban/>
[Accessed 11 12 2023].
- ionos, 2023. *about-kanban*. [Online]
Available at: <https://www.ionos.ca/digitalguide/websites/web-development/about-kanban/>
[Accessed 11 12 2023].

- Keita, B., 2023. *scrum methodology pros and cons*. [Online]
Available at: <https://www.invensislearning.com/blog/scrum-methodology-pros-and-cons/>
[Accessed 22 12 2023].
- kissflow, 2023. *kanban-methodology*. [Online]
Available at: <https://kissflow.com/project/agile/kanban-methodology/>
[Accessed 11 12 2023].
- Krupa, L., 2023. *scrum-testing*. [Online]
Available at: <https://testsigma.com/blog/scrum-testing/>
[Accessed 23 12 2023].
- lumitex, 2023. *Prototype*. [Online]
Available at: <https://www.lumitex.com/blog/prototyping-methodology>
[Accessed 21 11 2023].
- netsolutions, 2023. *food deley every app today time*. [Online]
Available at: <https://www.netsolutions.com/insights/top-10-successful-food-delivery-apps-in-the-world/>
[Accessed 9 september 2023].
- nodejs.org, 2023. *nodeJS*. [Online]
Available at: <https://nodejs.org/en/about>
[Accessed 29 12 2023].
- Paudel, S., 2023. *about bhojdeals*. [Online]
Available at: <https://www.nepalitelecom.com/best-food-delivery-apps-in-nepal>
[Accessed 22 12 2023].
- pubnub, 2023. *About Map API*. [Online]
Available at: <https://www.pubnub.com/guides/what-is-a-map-api/>
[Accessed 29 12 2023].
- S, B., 2023. *scrum-methodology*. [Online]
Available at: <https://www.nimblework.com/agile/scrum-methodology/>
[Accessed 23 12 2023].
- scrum.org, 2023. *scrum methodology*. [Online]
Available at: <https://www.scrum.org/resources/what-scrum-module>
[Accessed 22 12 2023].
- scrum.org, 2023. *what-is-a-sprint-in-scrum*. [Online]
Available at: <https://www.scrum.org/resources/what-is-a-sprint-in-scrum>
[Accessed 14 10 2023].
- simplilearn, 2023. *advantage of using scrum methodology for project*. [Online]
Available at: <https://www.simplilearn.com/scrum-project-management-article>
[Accessed 22 12 2023].
- simplilearn, 2023. *reactjs*. [Online]
Available at: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
[Accessed 25 12 2023].
- Srivastava, A., 2023. *prototype model*. [Online]
Available at: <https://www.codingninjas.com/studio/library/prototype-model-in-software-engineering>
[Accessed 11 12 2023].

tipsnepal, 2023. *About Foodmandu*. [Online]

Available at: <https://www.tipsnepal.com/foodmandu-case-study-the-secret-success-recipe/>

[Accessed 11 12 2023].

visualcrossing, 2023. *About weather api*. [Online]

Available at: <https://www.visualcrossing.com/resources/documentation/weather-api/what-is-a-weather-api/#:~:text=A%20weather%20API%20is%20an,%2C%20well%2Ddefined%20programming%20interface>

[Accessed 28 12 2023].

wrike.com, 2023. *Scrum guide*. [Online]

Available at: <https://www.wrike.com/scrum-guide/scrum-sprints/>

[Accessed 10 6 2023].

yourstory, 2022. *web application treand*. [Online]

Available at: <https://yourstory.com/2015/07/web-application>

[Accessed 27 september 2023].

7. Appendix

7.1 SRS Document

7.1.1 Introduction

a. Purpose

The Software Requirements Specification (SRS) document is created to provide detailed information about system functionalities and non-functionalities. This document specifies the features required for web applications. This document also includes detailed information about hardware, software, and other technical requirements.

b. Intended Audience

This document will serve as a blueprint for the system's developers as well as the tester. As a result, users can use this document to learn about the features that are required in web applications. testers can read this document to learn about the project's vision.

c. Project scope

The goal of this project is to create a web application for improving the food ordering experience by providing food and restaurant options according to user budgets and contextual recommendations based on real-time data such as weather conditions. Users will be able to browse menu items and customise their orders. FoodAaja will be made in a way that makes online food ordering more easy and more enjoyable to each user.

7.1.2 Overall Description

7.1.2.1 System Perspective

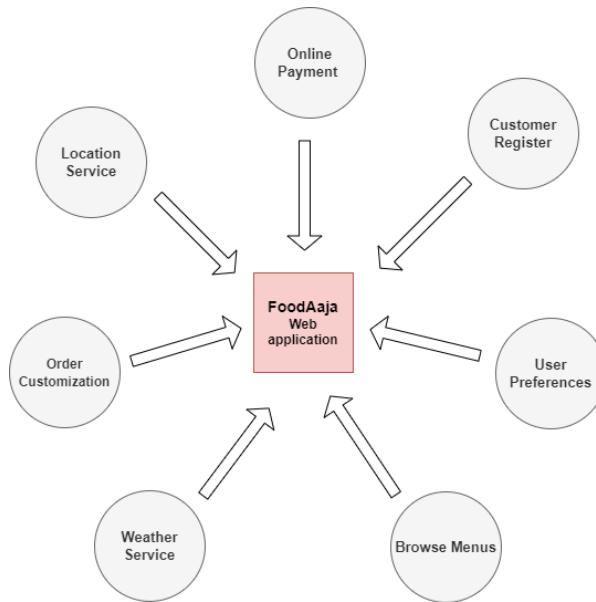


Figure 25: System Perspective of Web Application

7.1.2.2 System Features

Some of the features that should be included in a web application are listed below:

1) Register Customer

The web application will facilitate customers to register and create personal accounts. User needs to enter information such as their name, email, address, phone number, and password. After they have registered, they can update their profile information and change their password.

2) Food ordering

Customers can select a restaurant, choose their preferred dishes from the restaurant's menu, and add them to their cart. Once items are in the cart, they can proceed to checkout and payment. customers who have placed an order successfully can view the details of their order on the orders page, where they also have the option to cancel their order if needed.

3) Customer Preferences

The web application will facilitate customers to set budget ranges for meal selections and restaurants.

4) Browse Menus

The web application will facilitate customers and admins easy searching to find specific dishes or explore new option.

5) Recommendation Service

The system utilizes location data, time of day, and weather condition APIs to offer real-time, context-specific recommendations, such as suggesting warm soups on a cold day.

6) Order Customization

Customers can customize their meals while placing an order. they have the freedom to change the ingredients, specify how much they want to eat, and choose how it will be prepared. the system ensures that the eatery is fully aware of the user's preferences. customers can customize their meals while placing an order. they have the freedom to change the ingredients. the system ensures that the user gets exactly what they want.

7) Online Payment

Online Payment will be available for a normal user to pay the total cost of the order through Khalti.

8) Admin Panel

The web application will facilitate the admin to add new restaurants to the page and input their menu items. Admins can assign menus to specific restaurants, update them as needed, and manage various details like pricing and item availability. the admin can track all orders placed by customers.

7.1.3 Functional requirement

7.1.3.1 Customer Register

Req. ID	Requirement Description	
FR. 01	Sys. Req. ID	System Requirement
	SR.01	Customer can view the register form.
	SR.02	The system shall check whether the compulsory area is entered or not.
	SR.03	The system shall check whether the provided information is valid or not.
	SR.04	The system shall check user entered email is unique to this application or not.
	SR.05	The system shall display message whether the user registration is successful or not.

Table 2:Functional requirement description of customer register

7.1.3.2 Login

Req. ID	Requirement Description	
FR. 02	Sys. Req. ID	System Requirement
	SR.06	The system shall provide a login form that requests a username and password.
	SR.07	The system shall validate the entered username and password against the stored credentials.
	SR.08	The system shall display error message if user login is unsuccessful.
	SR.09	If user type is customer, then the system shall display customer dashboard otherwise admin dashboard should be display.

Table 3:Functional requirement description of login

7.1.3.3 Food ordering

Req. ID	Requirement Description
FR. 03	Customers can order food from a menu of a selected restaurant.
Sys. Req. ID	System Requirement
SR.10	The system shall provide a list of restaurants to the user based on their location and user ratings.
SR.11	If the customer click on a restaurant, then the system shall display the menu items of the restaurant with detailed descriptions and prices.
SR.12	If the customer click the add to cart button the system shell adds menu items to their order if he/she is logged in.
SR.13	If the customer clicks the Proceed to Order button then The system shall provide the option to select a current location on the map for delivery and the system shall provide an estimated total cost and quantity before order confirmation.
SR.14	If the customer click on proceed to payment then the system shall display an ordering confirmation message. If the customer confirms ordering, then the system shall display the payment page.
SR.15	The system shall offer online payment methods by Khalti and a pay-on delivery option for the user to choose from. if the customer click the Pay by Khalti option then Khalti UI open where a customer needs to enter the payment credentials and confirm payment. After payment the system shall display an ordering successful message.

Table 4:Functional requirement description of food ordering

7.1.3.4 Customer Profile

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
FR. 04	All the register customers can view their own profile.	
	SR.16	When customer click on profile button then the system shall display their profile details according to their email.
	SR.17	The system shall also allow the option to change private information like password.
	SR.18	The system shall check whether the user entered old password, new and confirmed password is valid or not.
	SR.19	If customer account password was changed successfully then display password changed message otherwise display error message.
	SR.20	The system will verify that the email and name entered by the user are valid and not already in use. If the username and email are not valid and are already used then it displays an error message.
	SR.21	If the customer successfully updates their profile, the system will display a success message.

Table 5: Functional requirement description of Profile

7.1.3.5 View Orders

Req. ID	Requirement Description	
FR. 05	Sys. Req. ID	System Requirement
	SR.22	When customer click on orders button then the system shall display their order details with their status.
	SR.23	If a customer clicks the cancel button, the system will let them cancel the order if it's within 30 minutes of ordering. It will also ask for confirmation before the order is finally cancelled.
	SR.24	The system shall display a confirmation message once an order is successfully cancelled or if the cancellation window has passed.

Table 6:Functional requirement description of orders

7.1.3.6 Recommendation service

Req. ID	Requirement Description	
FR. 06	Sys. Req. ID	System Requirement
	SR.25	The system shall integrate with weather condition APIs to obtain real-time weather data for the user's location.
	SR.26	The system shall integrate with a weather API to retrieve real-time weather data and use this information to tailor food suggestions (e.g., warm dishes on cold days, light meals on hot days).
	SR.27	The system shall consider the time of day in its recommendations, such as suggesting breakfast items in the morning, or dinner options in the evening.

Table 7:Functional requirement description of recommendation service

7.1.3.7 Admin Panel

Req. ID	Requirement Description	
	Sys. Req. ID	System Requirement
FR. 07	Admin users can manage restaurant listings, menu items, and view customer orders in the Admin Panel.	<p>SR.28 When the admin user click the add restaurants button the system shall allow admins to add new restaurants to the platform, including details such as restaurant name, location, and rating.</p> <p>SR.29 The system shall enable admins to remove existing restaurants from the platform.</p> <p>SR.30 The system shall allow admins to add and remove menu items from the particular restaurant.</p> <p>SR.31 The system shall enable the admin to edit the information of existing restaurants and their menu items, such as updating food descriptions and prices.</p> <p>SR.32 The system shall allow admins to view detailed information about each customer's order, including the order date, items ordered, total cost, and order status.</p>

Table 8: Functional requirement description of Admin Panel

7.1.4 External Interface Requirement

7.1.4.1 User Interface

The User Interface (UI) for our food ordering web applications will be designed and developed with user-friendliness to ensure ease of navigation. So that user won't be confused regarding pages, which they want to go with. Each page's functionality will determine the appropriate color scheme and the declaration of the page's name. the buttons and links are easy to find and use, and the layout will work well on both mobile and desktop devices.

7.1.4.2 Hardware

Customers must have a compatible hardware device, such as a smartphone, or computer, along with a internet connection to access and use the FoodAaja web application.

7.1.4.3 Software

Standard resources will be used to develop and maintain database of FoodAaja application. For developing the web application, visual studio is required for coding. Similarly, noSQL database is used in to store the data and information of users that are required for application. MongoDB is used to manage data.

7.1.5 Non-Functional requirements

7.1.5.1 Performance Requirements

Req.ID	Requirement Description	Priority
PR.01	It should take only 3 seconds to load the web application.	Must
PR.02	The application should show the customer dashboard, and customers can browse items without needing to log in first.	Must
PR.03	All features of the web application must function correctly across major web browsers.	Must

Table 9:Non-functional requirement description of performance requirements

7.1.5.2 Safety Requirements

Req.ID	Requirement Description	Priority
SR.01	It should implement user authentication.	Must
SR.02	The application must securely handle user data to prevent unauthorized access.	Must
SR.03	All personal information must be encrypted	Must

Table 10:Non-functional requirement description of safety requirements

Go back to the development to the date [Click here](#).

7.2 Wireframe

7.2.1 Home Page Screen

On the customer dashboard homepage, a list of restaurants is displayed with information such as name, location, and rating. At the top, there is a navigation bar with options where users can access their profile and cart. A search bar is there to quickly find specific restaurants. A price filter is also there that allows the customer to select a range that fits their budget, and there is a separate section for recommended services.

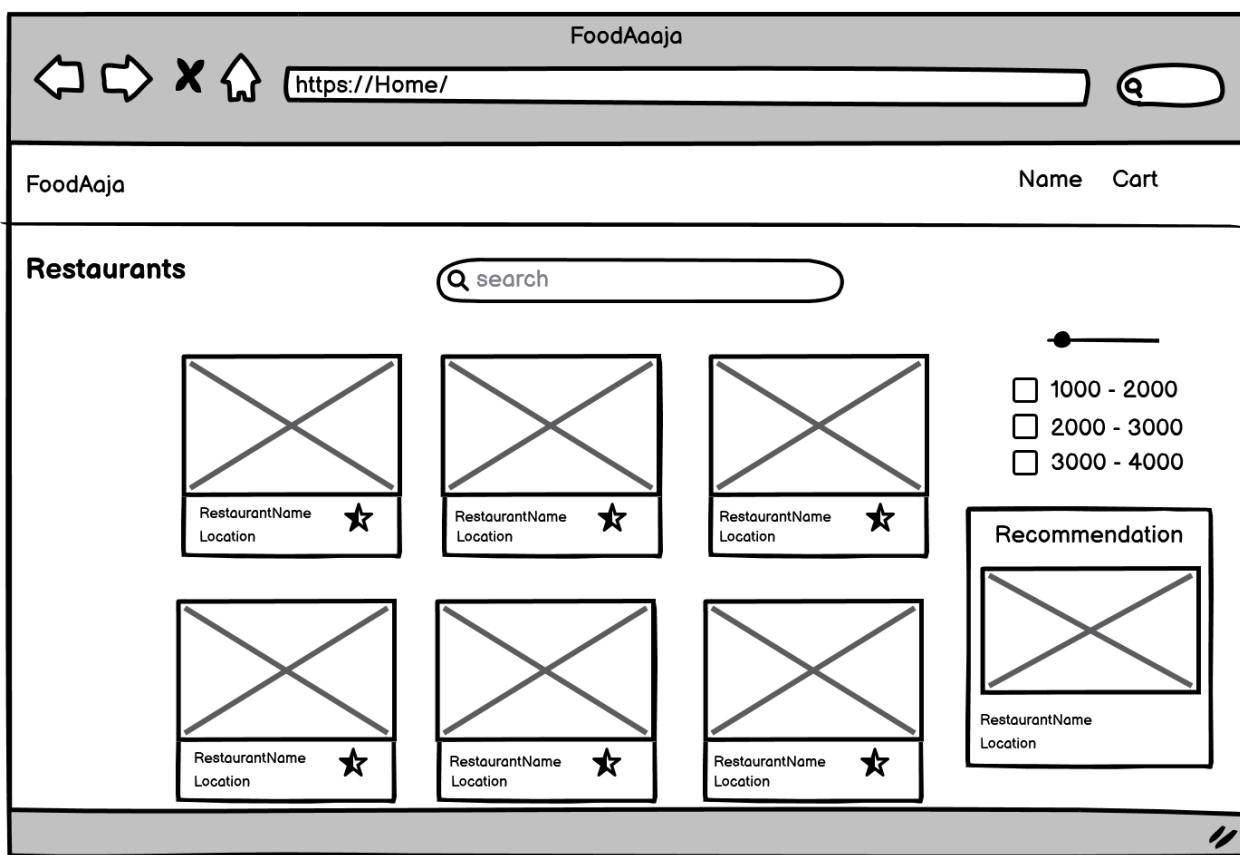


Figure 26: Wireframe of home page

7.2.2 Menu Page Screen

On the Menu Page, a list of specific restaurant menus is displayed with each food item's information such as name, price, and description and there is an add to card button also. a search bar is there to quickly find specific food items. A price filter is also there that allows the customer to select food items that fits their budget, and there is a separate section for food-recommended services.

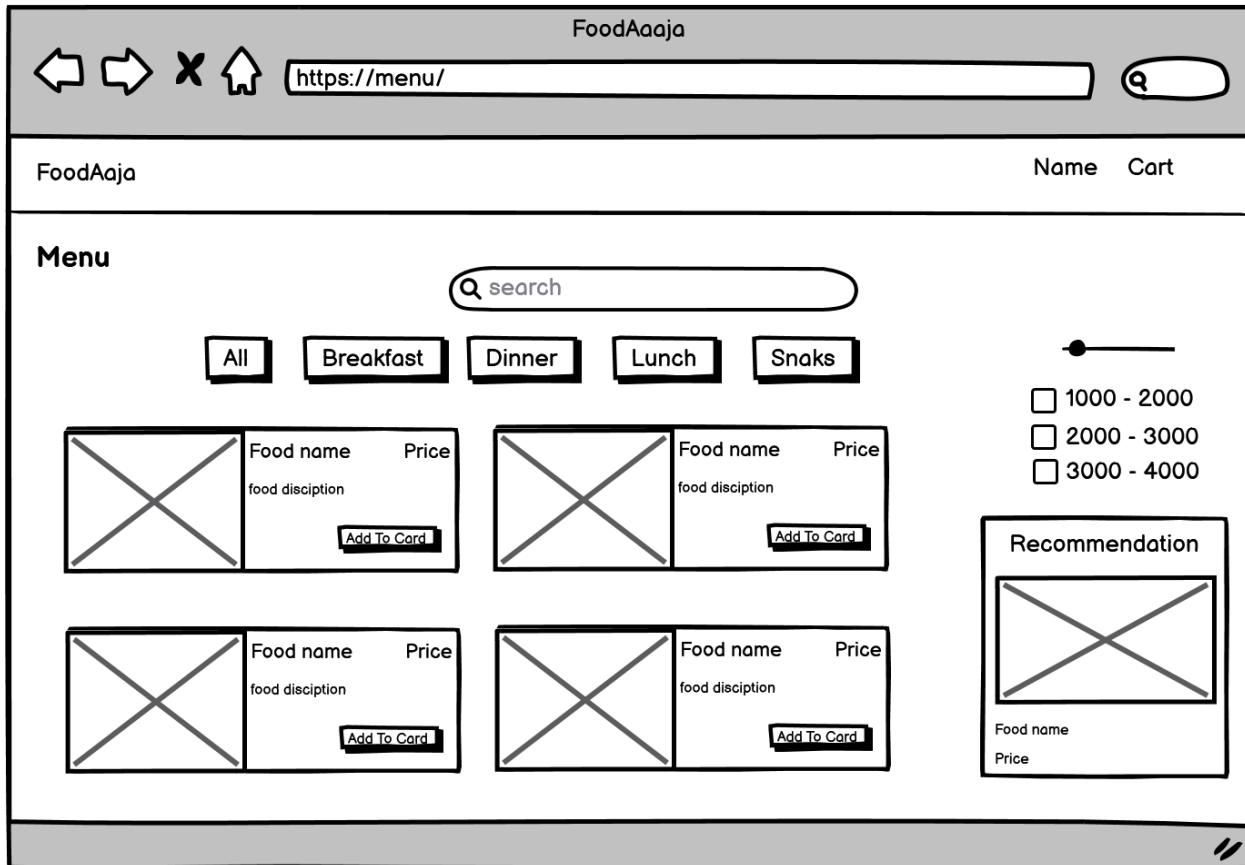


Figure 27: Wireframe of menu page of particular restaurant

7.2.3 Login Screen

This is the login screen for both users i.e. Customer and Admin. From this screen user enter their registered account email and password to use the features of foodAaja web application. If user entered email and password is valid then system checks whether that user is admin or customer. If user is customer, then it will display customer dashboard otherwise admin dashboard. Similarly, in this screen there is a Create your Account button for individuals who do not yet have an account, directing them to the registration screen.

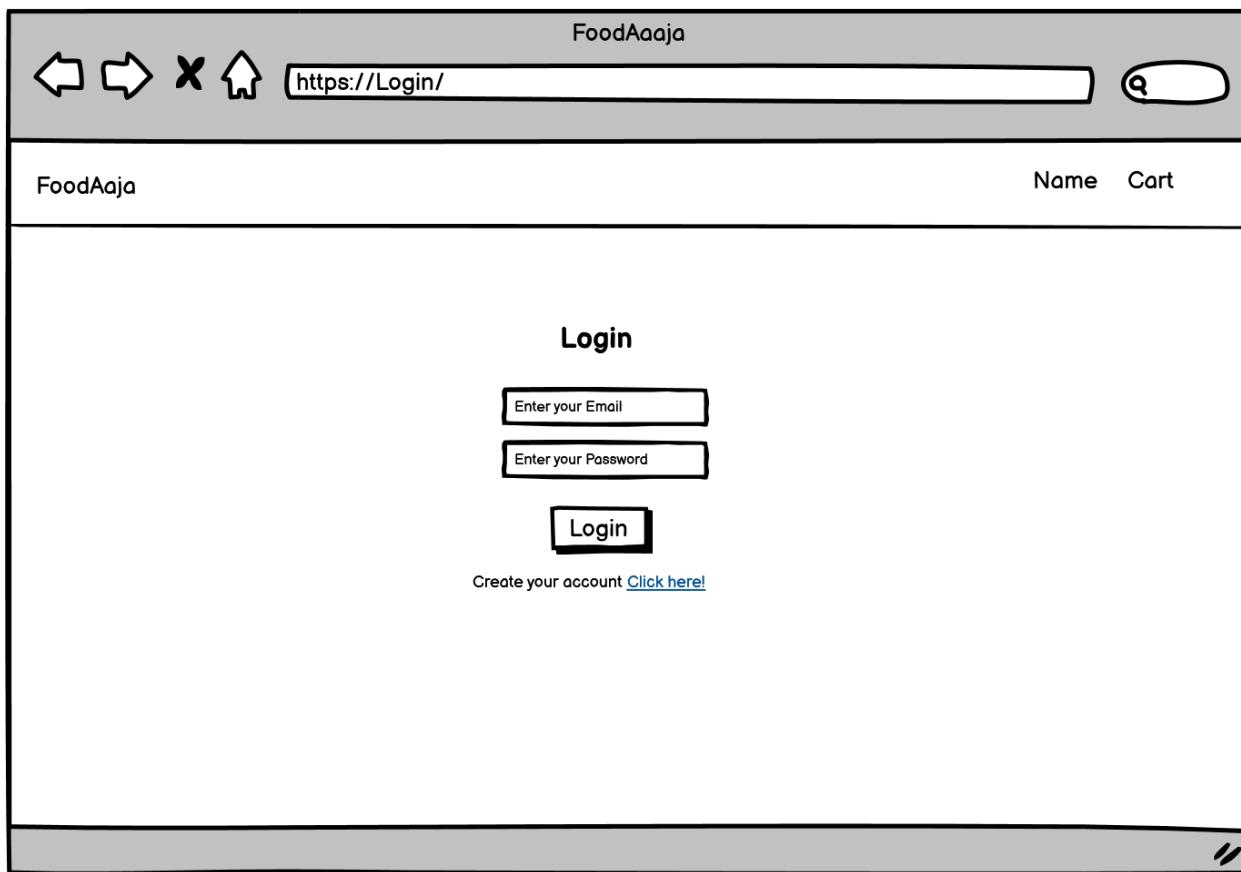


Figure 28: Wireframe of Login Page

7.2.4 Register Screen

This is the register screen for customers. If any customer want to use all the features of the application, a user must first register by filling out the given form. If a customer is already registered, they can navigate to the login screen by clicking the Already Have Account link.

The wireframe shows a web browser window for 'FoodAaja' at the URL <https://Register/>. The header includes standard navigation icons (back, forward, stop, home) and a search bar. Below the header, the page title 'FoodAaja' is displayed, along with links for 'Name' and 'Cart'. The main content area is titled 'Register' and contains five input fields for 'Full name', 'Address', 'Phone no.', 'Email', and 'Password', each with a placeholder text. A large 'Submit' button is centered below the inputs. At the bottom of the page, there is a link 'Already have account [Login](#)'.

Figure 29: Wireframe of register page

7.2.5 User Account Dropdown menu

In the navigation bar, there's a "Name" section which shows the users name. when we click it, it opens a dropdown menu with options like 'Profile', 'Orders', and 'Logout'.

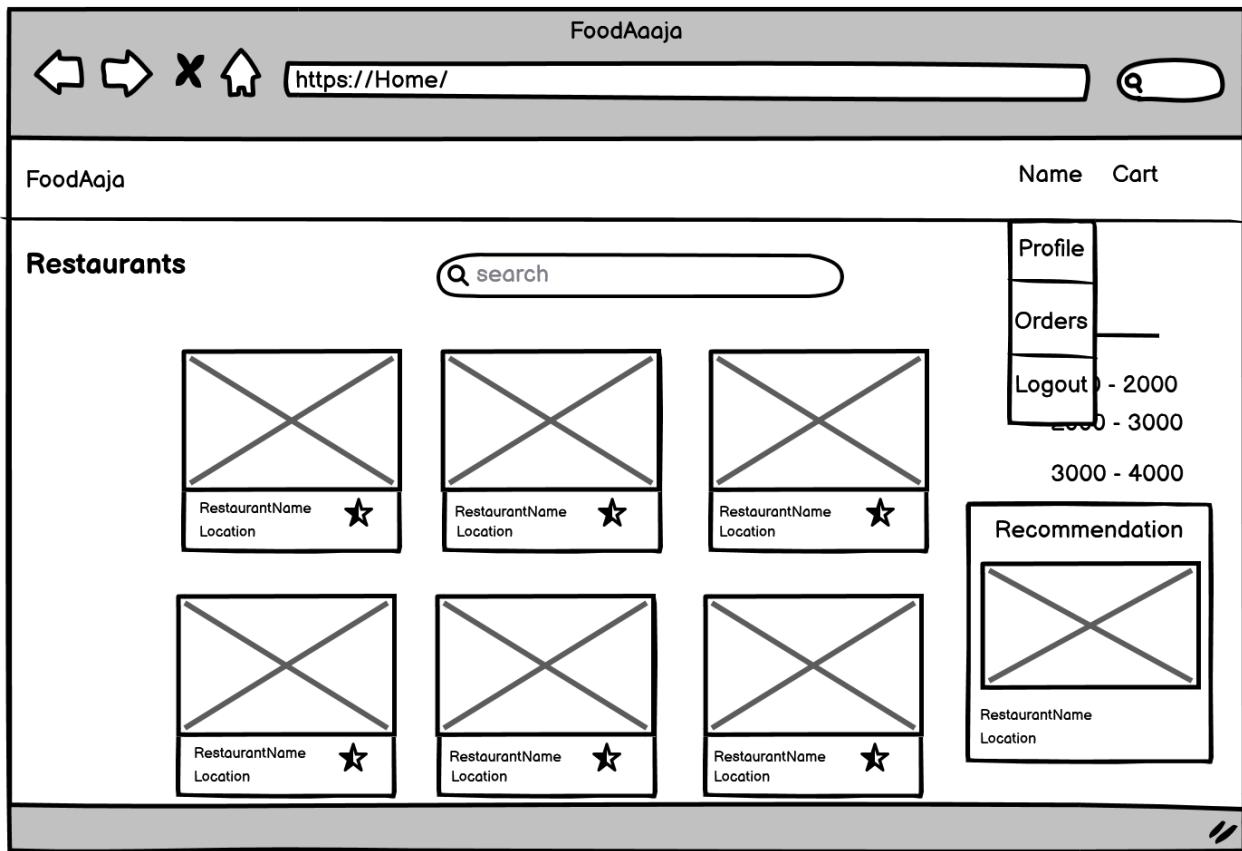


Figure 30: User Account Dropdown menu

7.2.6 Cart page screen

This is the cart page screen of the FoodAaja web application, where user can review the items they have added to their card for purchase. Each item is displayed with an image, food name, price, and a dropdown option for quantity. There is an input field for each item where user can input customization details to customized their food, and there is a remove button to delete items from the cart. A summary box displays the total item count and total price, with a “process to checkout” button that redirects users to the checkout page.

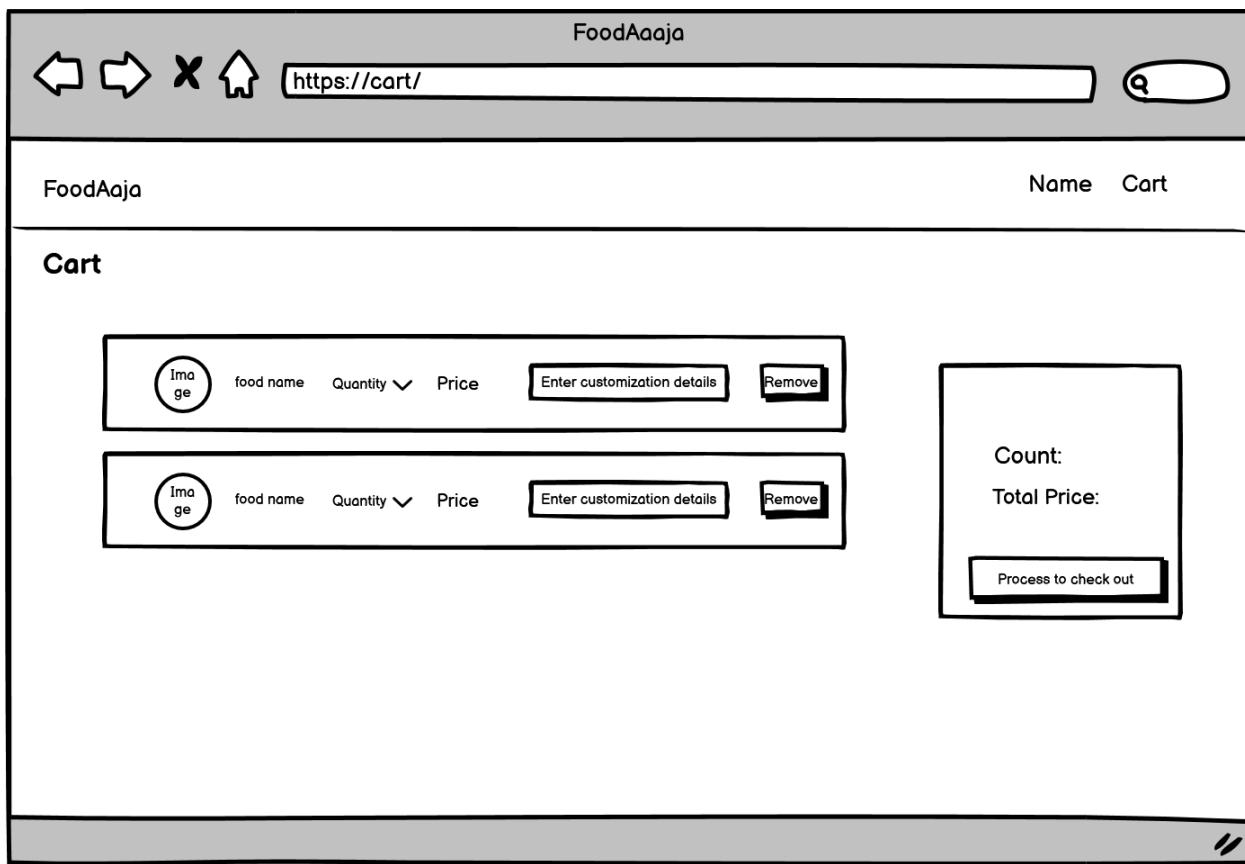


Figure 31: Wireframe of cart page

7.2.7 Check out page screen

This is the checkout page screen of the application, it is developed for finalizing orders. Users can provide order details such as name and address in the field provided. The page displays a summary list of each item with image, name, single item price, total quantity, and total price with any customization details. There is also a section for customer to choose their delivery location with a map with the “Find My Location” functionality. When the customer click the “Go to payment” button he/she redirects to the payment page.

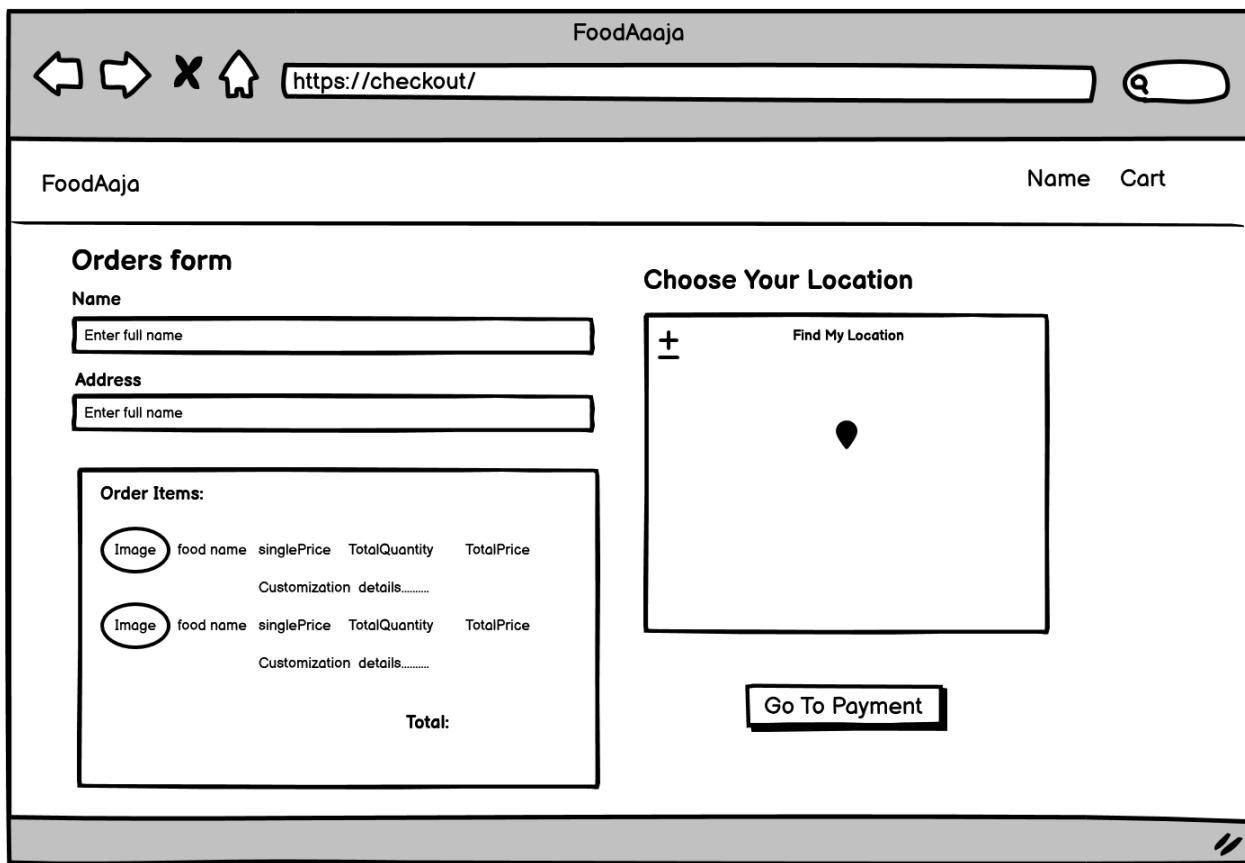


Figure 32: Wireframe of checkout page

7.2.8 Payment page screen

This is the Payment page screen of the application, the user order details are summarized for payment. It displays the username and address of a customer and a list of ordered items with their details. To the right, there is a map indicating the customer's selected location. The total cost of orders is displayed. There are two payment options, "Pay by Khalti" for online payment and "Pay on Delivery" for cash payment. When a customer click the "Pay by Khalti" button he/she redirects to Khalti payment UI.

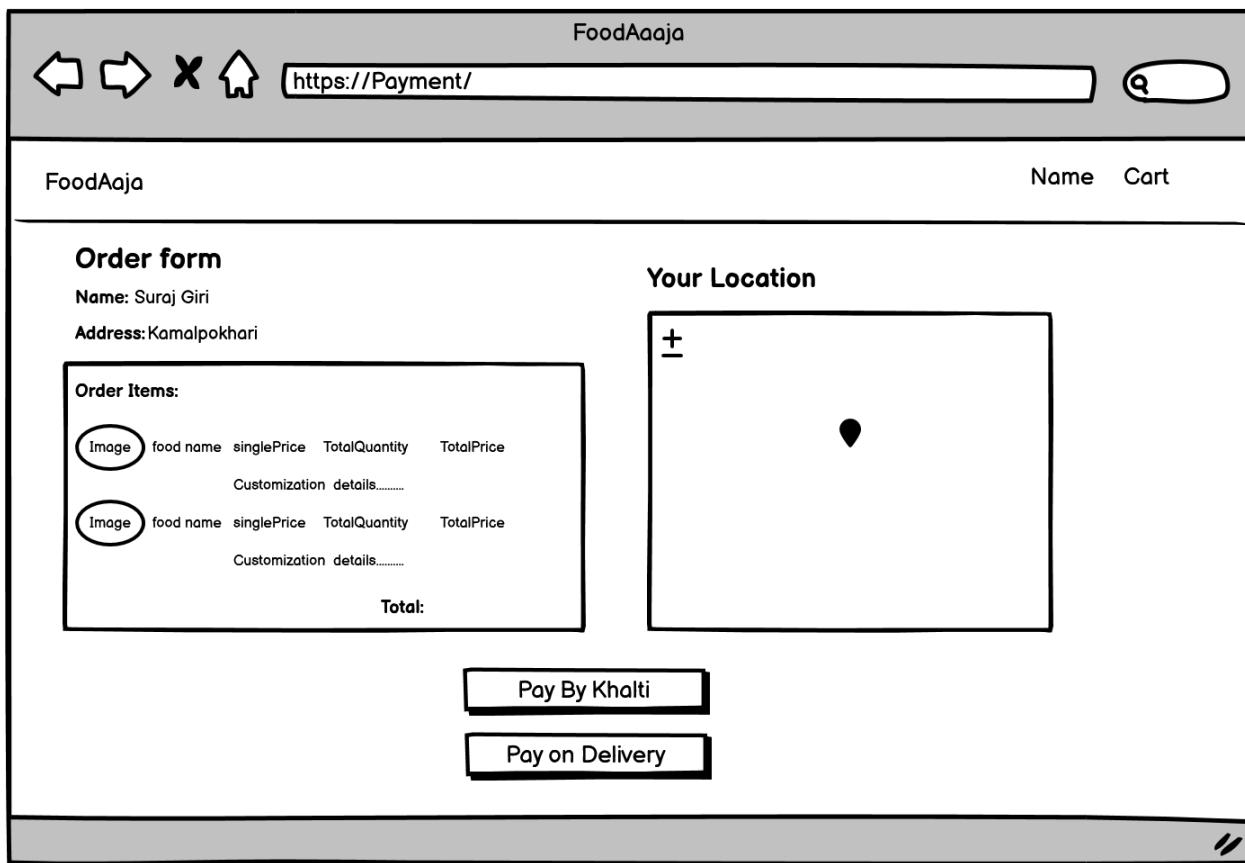


Figure 33: Wireframe of Payment page

7.2.9 Profile page screen

From the customer profile screen, customer can update their personal information, such as full name, address, and phone number, with pre-filled fields showing existing details. Customers can update their password by entering their current password,new password and confirm password. "Update" button will update the profile and "Change' button will change password.

The wireframe illustrates a web-based profile management interface. At the top, there's a header bar with the logo 'FoodAaja', navigation icons (back, forward, search), and a URL bar showing 'https://Profile/'. Below the header, the main content area has a header 'FoodAaja' on the left and user links 'Name' and 'Cart' on the right. The central part of the page is divided into two main sections:

- Update Profile**: This section contains three input fields: 'Full name:' with 'Suraj giri' entered, 'Address:' with 'Kamalpokhari' entered, and 'Phone no:' with '9800000000' entered. Below these fields is a large 'Update' button.
- Change Password**: This section contains three input fields: 'Current Password' (placeholder 'Current Password'), 'New Password' (placeholder 'New Name'), and 'Confirm Password' (placeholder 'Confirm Password'). Below these fields is a large 'Change' button.

Figure 34: Wireframe of profile page

7.2.10 Orders page screen

On the customer orders screen, customers can view their order history. Each order is listed with its name, image, price, Order ID, date and time of the order, and payment status. Customers can filter orders by clicking on tabs like "All", "Paid", "Unpaid", and "Canceled". If customers click the 'Cancel Order' button, the order will be canceled if it is within 30 minutes of being placed.

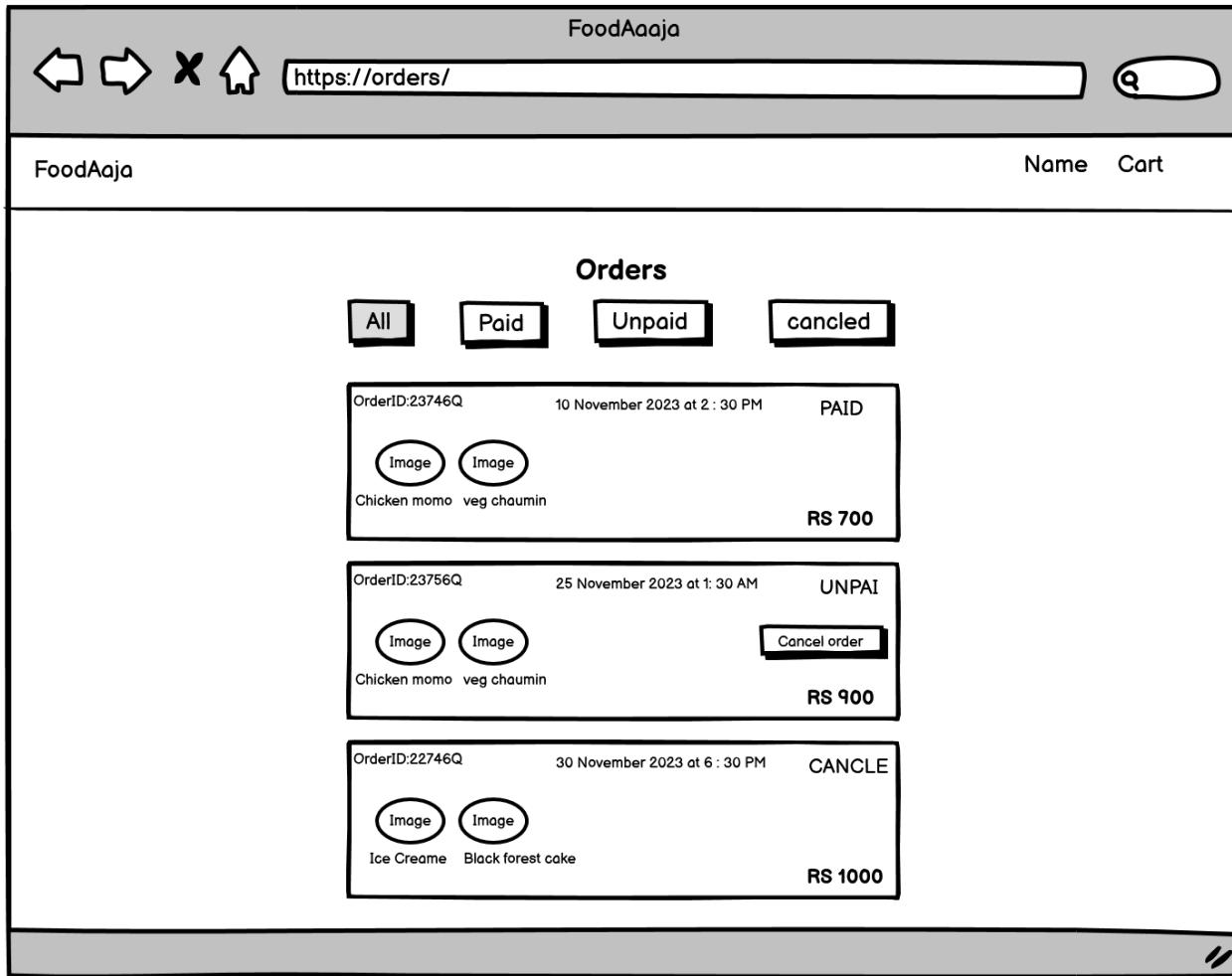


Figure 35: Wireframe of orders page

7.2.11 Admin dashboard page screen

The Admin dashboard page screen is designed for managing admin part. It displays a vertical navbar menu with options for "Restaurant", "Orders", and "Logout". There is a search bar for finding specific restaurants quickly. Users can edit restaurant information by clicking the "Edit" button, or users can remove a restaurant from the list by clicking the "Remove" button. Users can also add a restaurant by clicking the "Add Restaurant" button.

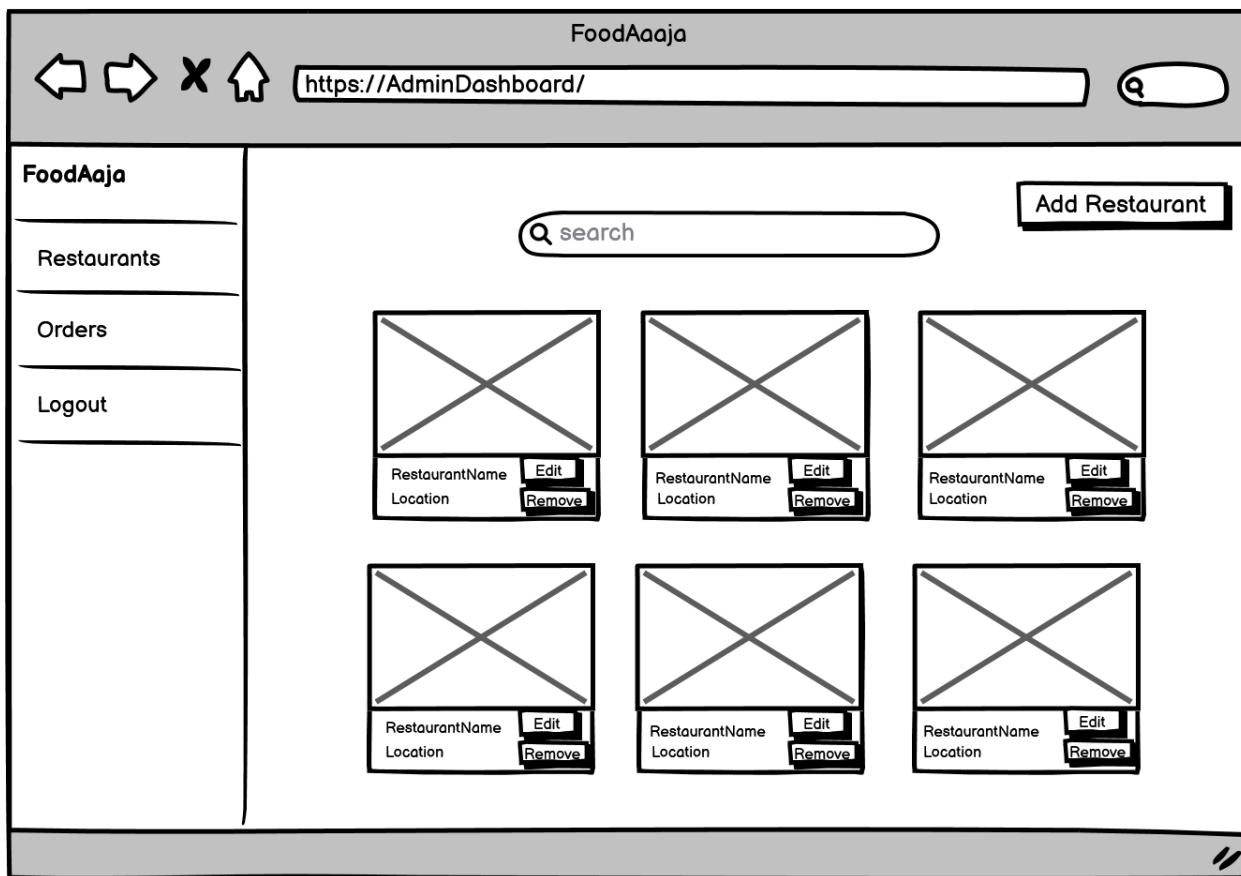


Figure 36: Wireframe of the admin dashboard

7.2.12 Add restaurant screen

It displays the add restaurant feature within the admin dashboard. It shows the form where new restaurant details can be entered. An admin can enter new restaurant details like name, location, rating, stars, and image URL.

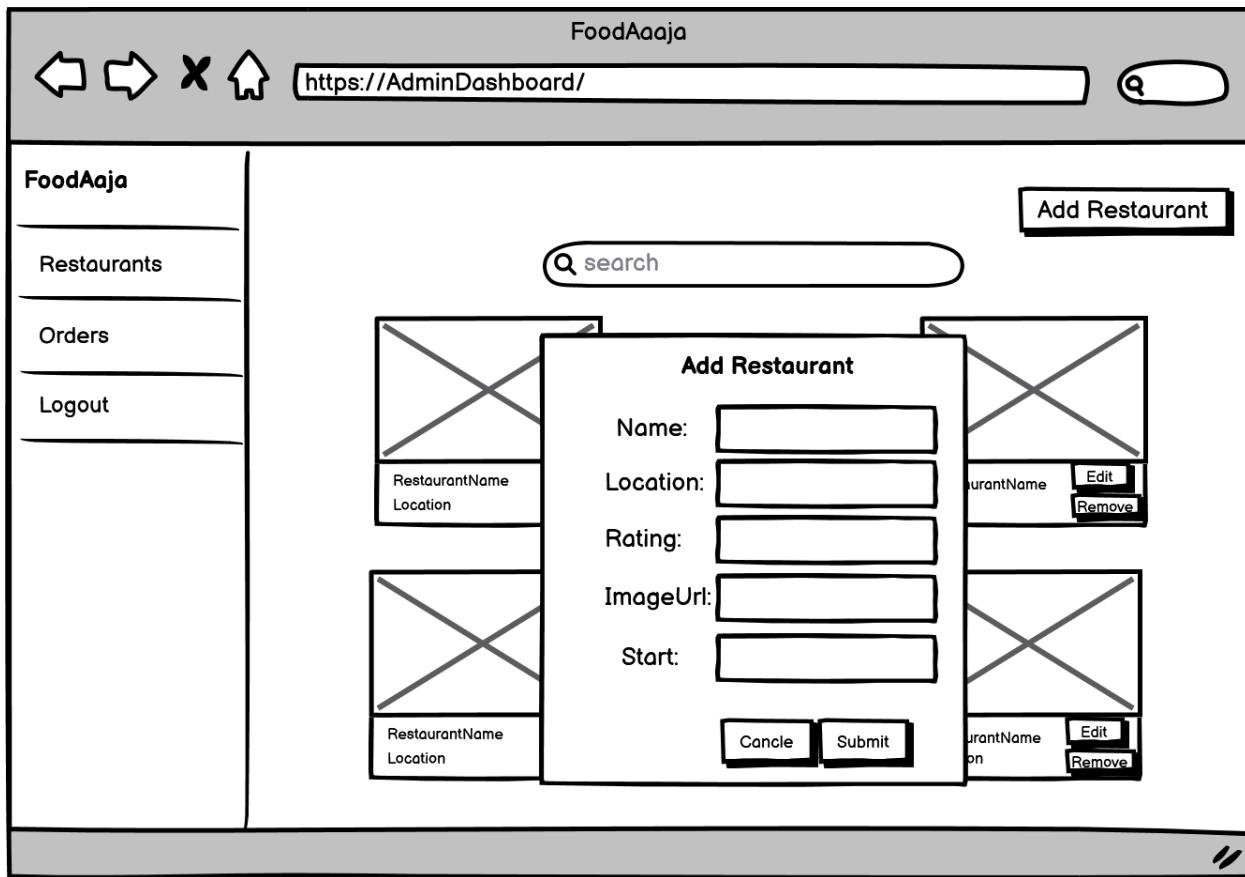


Figure 37: Wireframe of add restaurant

7.2.13 Menu management page screen

The menu management page screen is designed for managing menu items of the particular restaurant. There is a search bar for finding specific food items quickly. Users can edit food information by clicking the “Edit” button, or users can remove a food item from the list by clicking the “Remove” button. Users can also add a food item by clicking the “Add Food ” button.

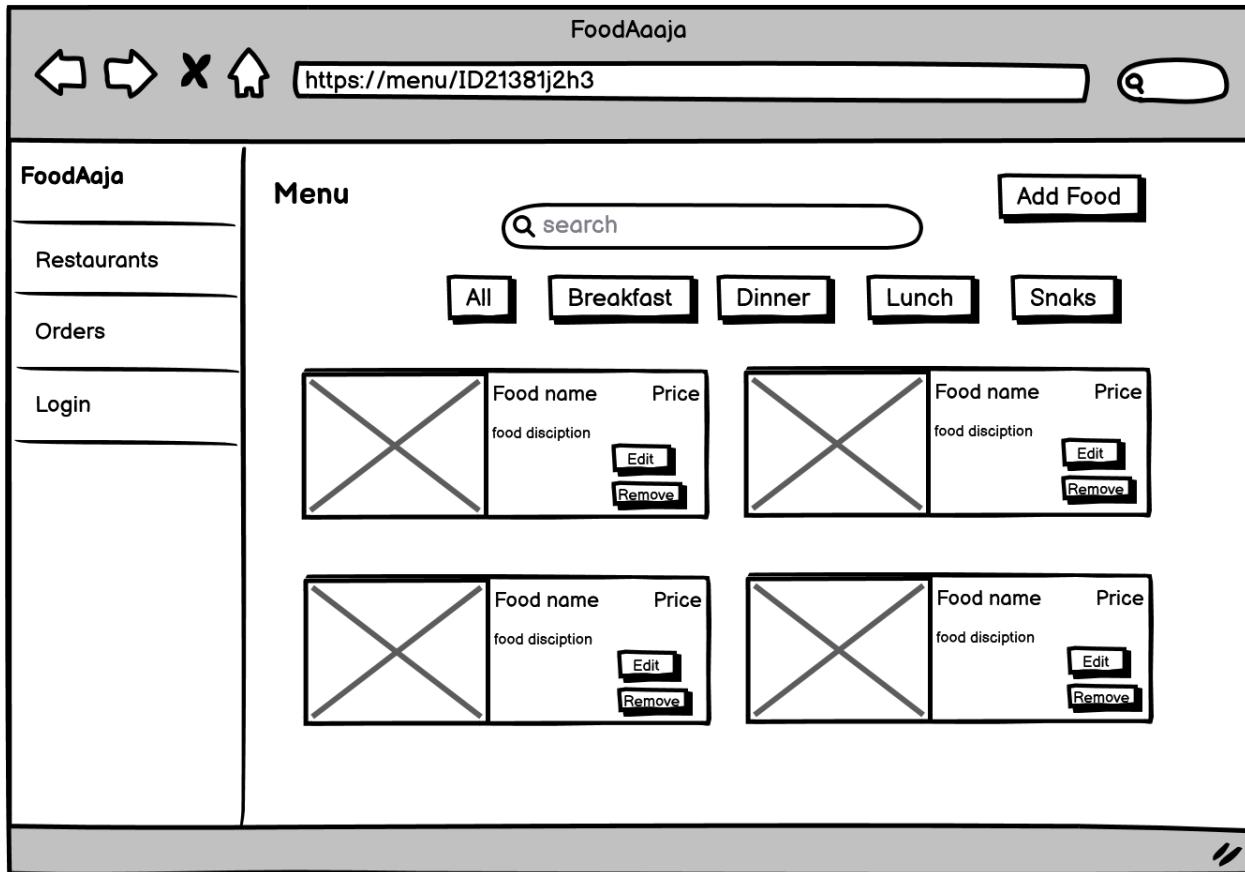


Figure 38: Wireframe of food management page

7.2.14 Add Food menu screen

It displays the add food feature within the admin dashboard. It shows the form where new food item details can be entered. An admin can enter new food details like name, price, tags, stars, description, and image URL.

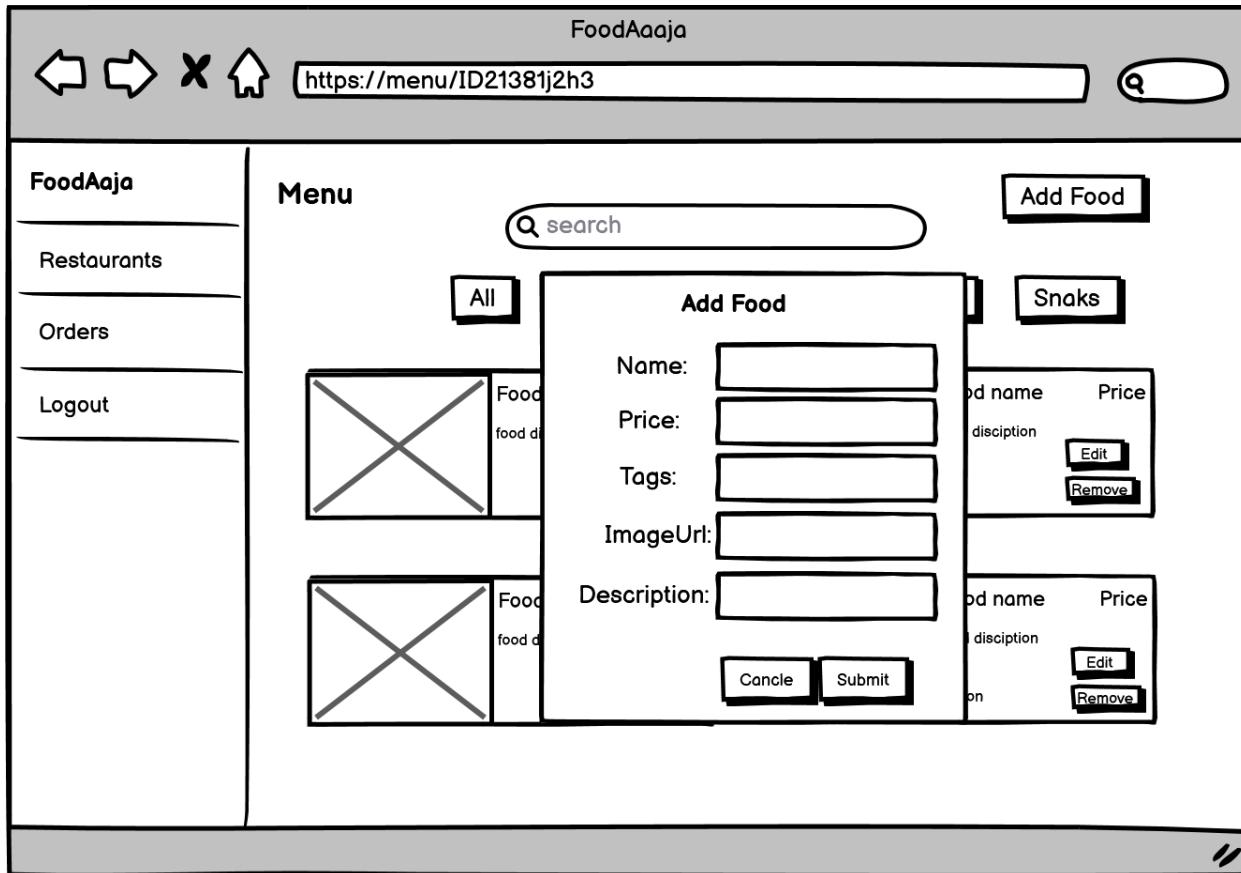


Figure 39: Wireframe of add food menu.

7.2.15 Ordered list page screen

The wireframe displays a list of orders placed by customers, including details such as the customer's name, address, phone number, order details, and the status of each order.

The wireframe shows a web browser window for 'FoodAaja'. The header includes back, forward, and search buttons, and a URL bar with 'https://Orders/'. The main content area has a sidebar with 'FoodAaja' logo, 'Restaurants', 'Orders', and 'Logout' links. The main content area is titled 'Orders' and contains a table with four rows of order data. The table columns are Name, Address, Phone, Orders, and Status.

Name	Address	Phone	Orders	Status
Giacomo Guilizz	Kamalpokha	980000640	Chicken momo chicken chaumin	PAYED
Hari sarma	Chabhil	980040064	Chicken momo chocolate Ice crea	UNPAYE
Karan karki	Kalopul	980550004	Veg momo Buff chaumin	UNPAYE
Dawa sherpa	Bouddha	980666004	Vanilla cake Apple pie	CANCEL

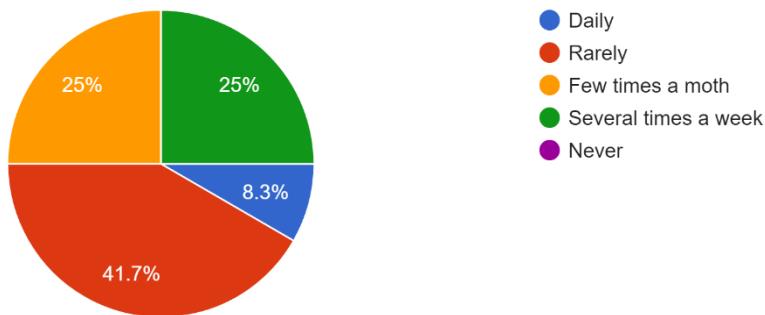
Figure 40: wireframe of admin orders listing page

Go back to the development to the data [Click here](#).

7.3 Survey Report

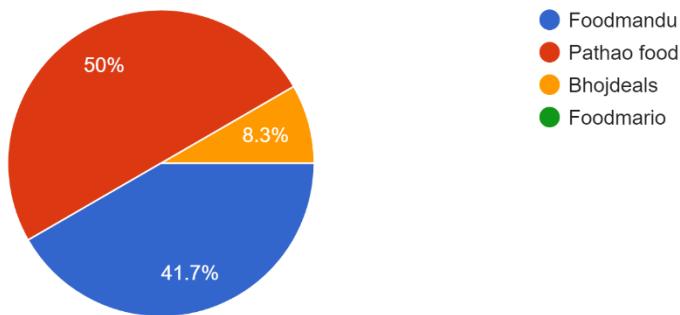
How often do you use online food ordering services?

12 responses



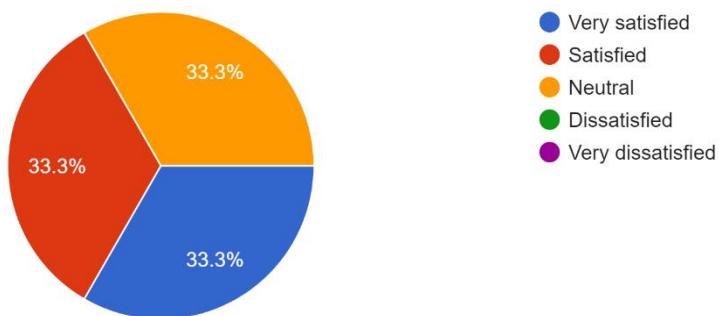
Which online food ordering platforms do you use most frequently?

12 responses



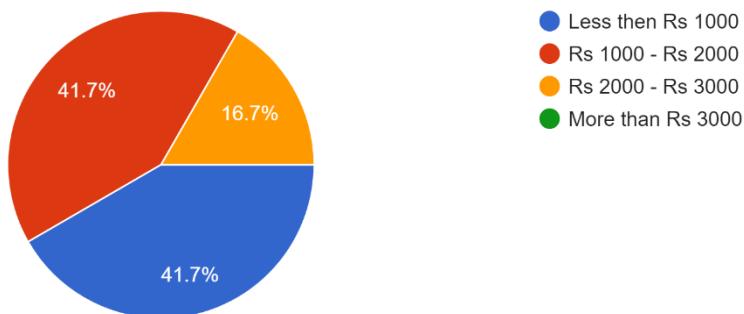
How satisfied are you with the online food ordering services you use?

12 responses



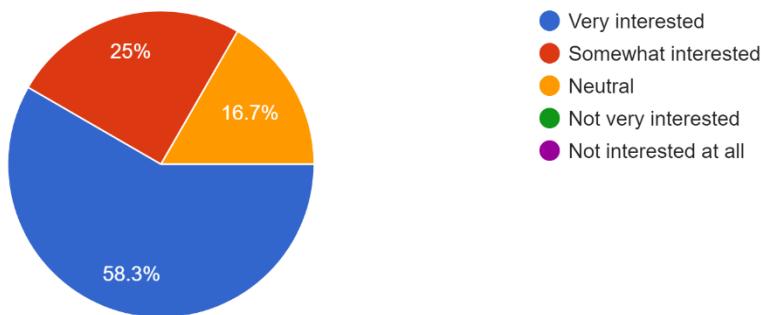
On average, how much do you spend per order when using online food ordering services?

12 responses



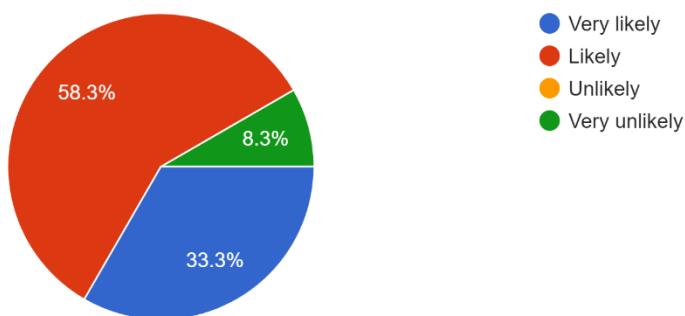
How interested would you be in a feature that recommends menu items or restaurants based on your specified budget?

12 responses



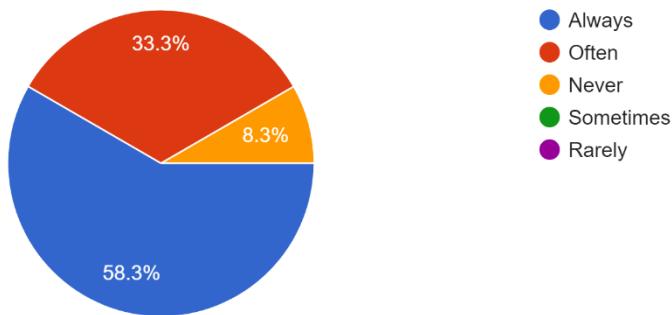
How likely would you be to use a budget-based recommendation feature on a food ordering platform?

12 responses



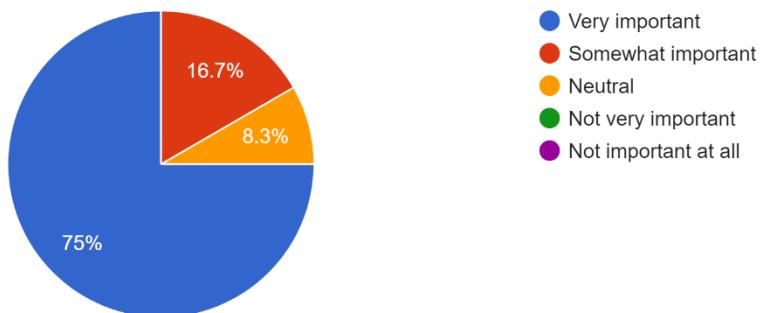
How often do you receive exactly what you ordered (correct items, portions, and customization)?

12 responses



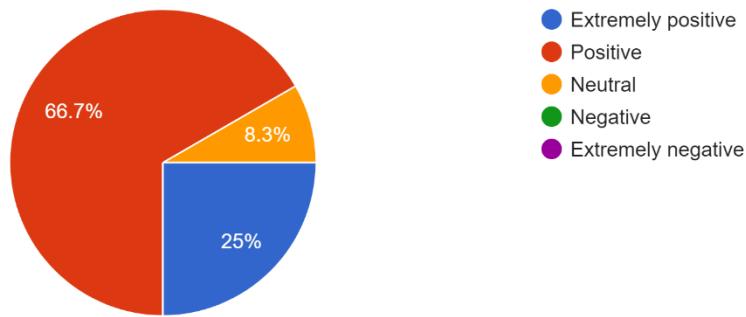
How important is the ability to customize your order (e.g., adding/removing ingredients) when you use online food ordering services?

12 responses



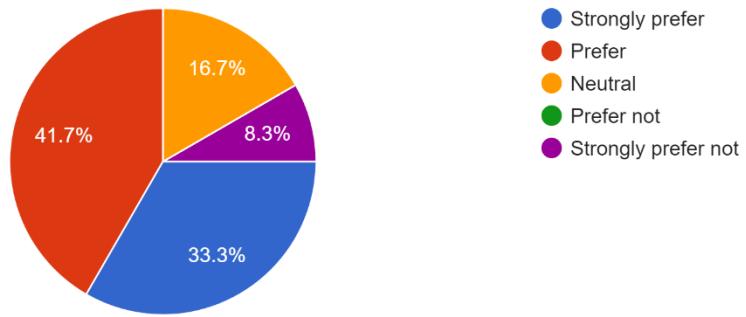
How would you rate your experience if a food ordering webapp suggested warm soups to you on a cold day?

12 responses



Do you prefer a food ordering app that provides personalized suggestions based on the weather or other contextual factors?

12 responses



Go back to the development to the date [Click here](#).

7.4 Considered Methodology

7.4.1 Prototype

The Prototype Methodology in software development involves creating an initial version of the final product, called a prototype, which is then tested and modified as needed until it meets the required standards. This method is particularly effective for projects with unclear or evolving requirements. It allows for a deeper understanding of user needs and the project's viability from an early stage (lumitex, 2023). By involving users in testing the prototype early on, their feedback helps in refining and enhancing the final product. This cycle of building, testing, and improving continues until the product fulfills the desired specifications, significantly reducing the risk of project failure due to misinterpreted or changing requirements (Srivastava, 2023).

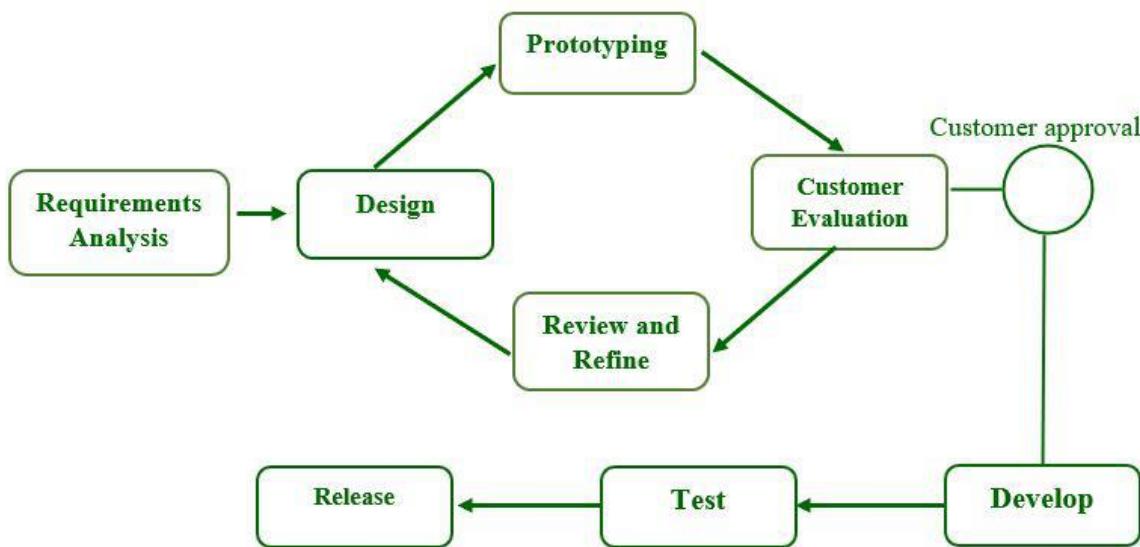


Figure 41: Prototype Methodology (geeksforgeeks, 2023).

7.4.2 Kanban

Kanban is a streamlined and agile method for software development that uses a visual board, called a Kanban board, to show the workflow process. This board helps teams track their tasks from start to finish. The main ideas of Kanban are to keep the amount of work manageable, focus on finishing tasks, and regularly improve the workflow (kissflow, 2023). Tasks are displayed as cards on the board, moving through different stages like "To Do," "In Progress," and "Done." This setup helps teams quickly spot and fix any delays or issues, leading to quicker completion times and a more adaptable approach to changing work needs. Kanban is especially useful when project requirements often change, offering great flexibility (ionos, 2023).

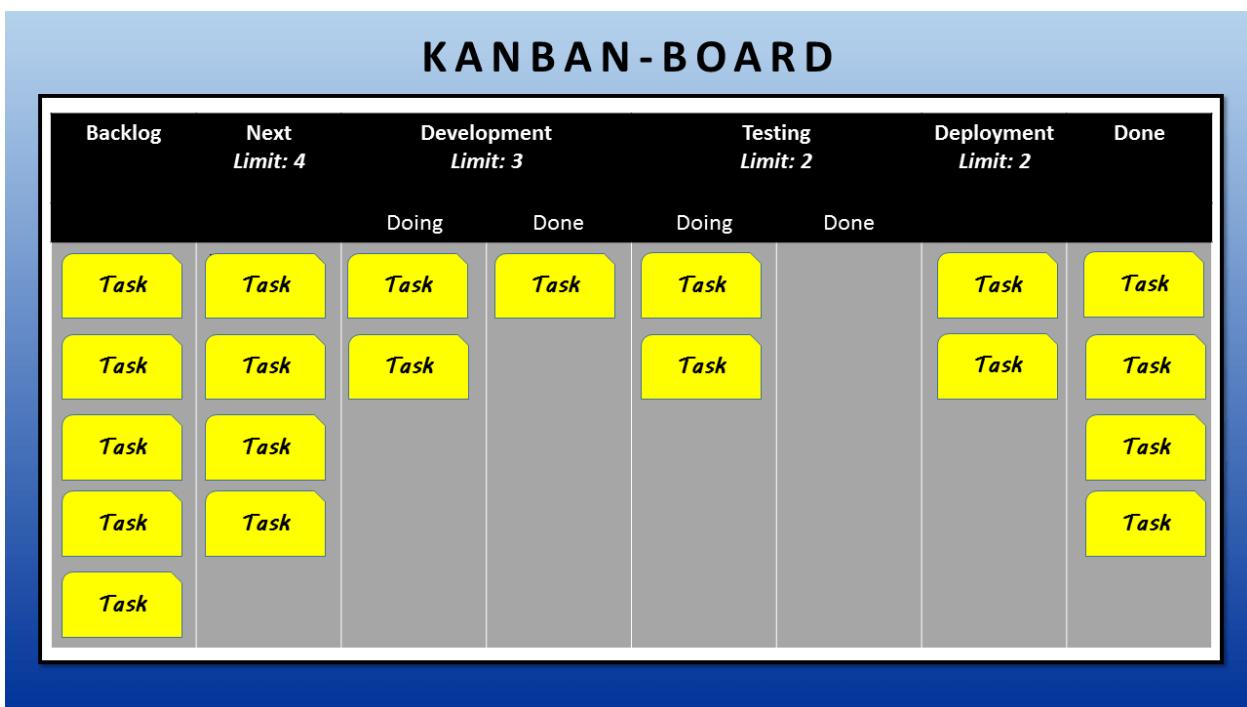


Figure 42: Kanban Methodology (ionos, 2023).

7.5 Selected Methodology

7.5.1 Scrum methodology

In this project, I have implemented Scrum methodology. Scrum is an agile project management framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks and practices that help teams in organizing and managing their work (atlassian, 2023). It is a way for teams to work together to develop a product. But it is not limited to team only, it works well for individual projects also. Personal Scrum is an Agile methodology that adapts and applies Scrum practices to one-person projects. It promotes personal productivity through observation, adaptation, progressive elaboration, prioritizing and sizing work, and time-boxing (infoq.com, 2023).

A Sprint is a time-boxed event of one month or less during which a potentially releasable product increment is created.

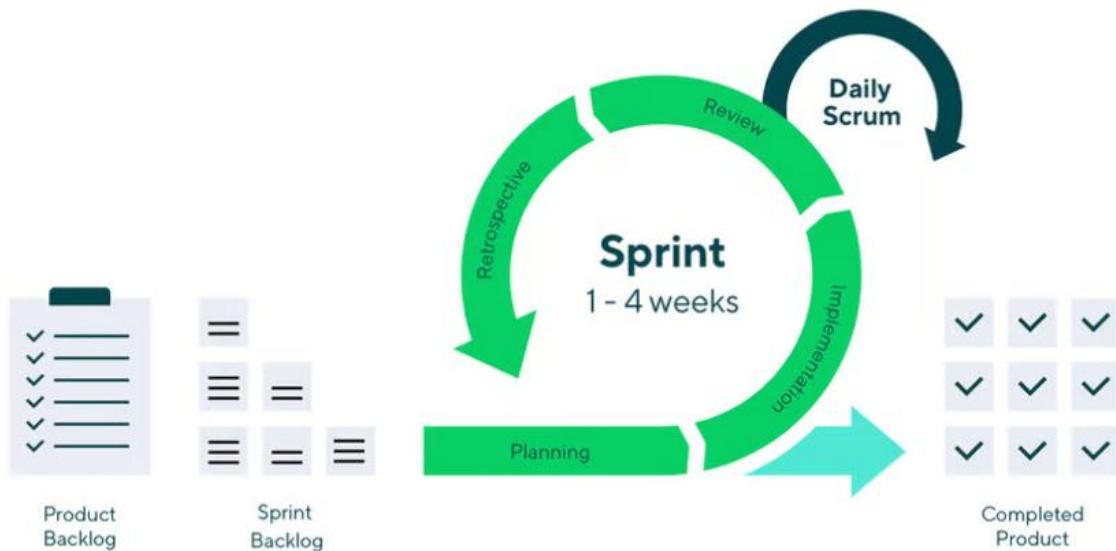


Figure 43: Scrum Methodology (wrike.com, 2023).

A new Sprint starts immediately after the conclusion of the previous Sprint 1. During the Sprint, no changes are made that would endanger the Sprint Goal, quality does not decrease, and the Product Backlog is refined as needed (scrum.org, 2023).

While originally designed for large, collaborative projects, many aspects of Scrum can be effectively adapted for individual projects. Scrum is applicable to our personal lives. The whole point is, through a process of constant self-awareness, to identify what is holding us back, how we can work around it, and where the next few days or weeks should take us.

Go back to the development to the date [Click here.](#)

7.6 Similar Project Description

a) FoodMandu

Foodmandu is a popular web app for ordering food online in Nepal. It allows people to order food from a variety of restaurants and have it delivered to their home. The app is simple to use and offers a wide range of food options. It was the first in Nepal to do so, transforming the way people order food. Foodmandu makes it easier and more convenient to order food (tipsnepal, 2023). Its features include real-time order tracking, a variety of cuisine options, and secure payment methods, all of which improve the ease and convenience of online food ordering (ekbana, 2023).

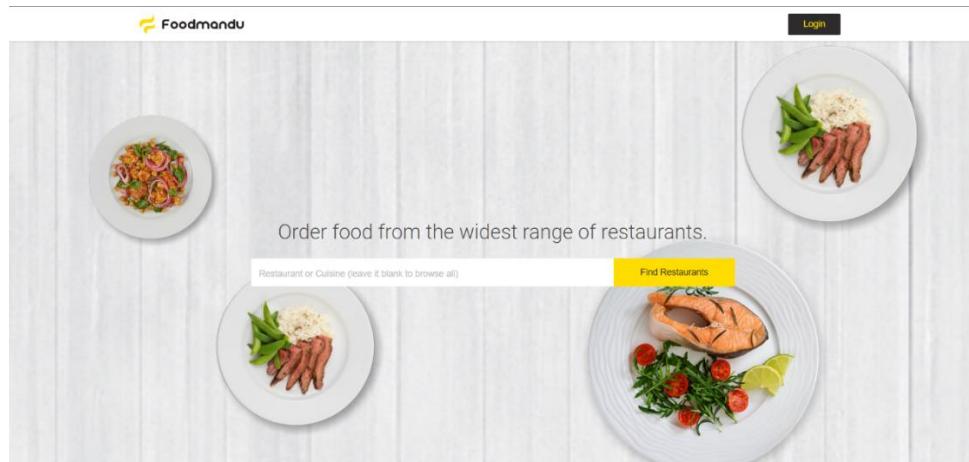


Figure 44: Foodmandu web application user dashboard

b) Bhoj

Bhoj, also known as Bhojdeals, is a food startup that has been delivering food online since 2015. It is a one-stop mobile application for all Nepalese foodies. Bhoj-Wallet is an excellent payment option in this app. Regular customers can use this service to get amazing deals and discounts from their favorite restaurants. These specials change every day. BhojDeals covers deliveries in major areas like Kathmandu, Bhaktapur, and Lalitpur. It accepts payments through fonepay, eSewa, credit/debit card, IPS, khallti , Paypal and also accepts cash on delivery. However, the map identifying services aren't as accurate most of the time. Its user-friendly interface simplifies the ordering process. it currently has a 2.9/5 rating and has received over 100k downloads (Paudel, 2023).

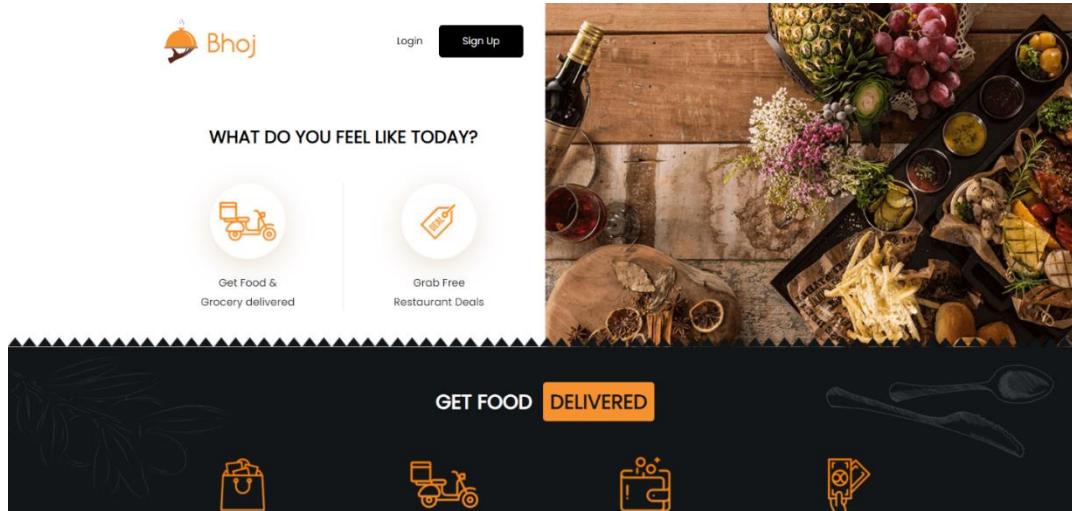


Figure 45: Bhojdeals web application user dashboard

c) Bhok Lagyo

Bhok Lagyo is a modern Nepalese food ordering system designed to meet the changing needs of Nepalese customers. It's simple to use and allows people to select and order food from a variety of local restaurants. The menu includes both Nepalese and international dishes. It has convenient features such as real-time order tracking, multiple payment options, and fast, dependable delivery. Bhok Lagyo wants to make eating out more enjoyable for its customers. It also benefits local restaurants by increasing their visibility to more customers, thereby promoting the growth of the local food industry (Ghimire, 2023).

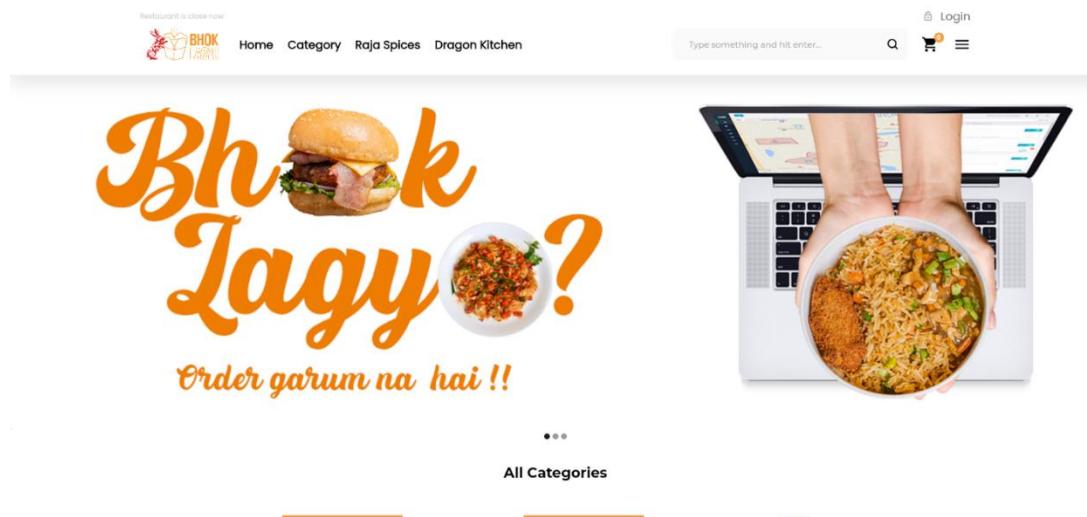


Figure 46: Bhok lagyo web application user dashboard

Go back to the review of existing system section [Click here](#)

7.7 Progress Table

SN	Tasks	Status	Progress (%)
1.	Research about topic in depth	Completed	100%
2.	Finalize the Project Topic	Completed	100%
3.	Look for similar project	Completed	100%
4.	Review proposal and Update proposal	Completed	100%
5.	Submit proposal	Completed	100%
6.	Gathering the required feature, resource and requirement	Completed	100%
7.	System architecture design	Completed	100%
8.	User Interface design	Completed	100%
9.	Wireframe and prototype	Completed	100%
10.	Database design	Completed	100%

Sprint 1

11.	Set up server infrastructure	Completed	100%
12.	Database connection	Completed	100%
13.	Login Backend/Frontend implementation	Completed	100%
14.	Authentication and Authorization	Completed	100%
15.	Add, Edit, Remove Restaurant and Menu Item Features	Partially Completed	40%
16.	Restaurant Listing Page	Partially Completed	40%
17.	Menu Items Listing Page	Partially Completed	40%
18.	Sprint Testing	Not Completed	0%

Sprint 2

19.	Add To Cart Functionality	Not Completed	40%
20.	Checkout feature	Not Completed	10%
21.	Payment Features with Khalti Pay integration	Not Completed	0%
22.	Map API Implementation	Not Completed	0%
23.	Sprint Testing	Not Completed	0%

Sprint 3

24.	Register, Orders and Profile Page	Not Completed	0%
25.	Backend Schema for user register	Not Completed	0%
26.	Edit profile and change password Functionality	Not Completed	0%
27.	Weather API Implementation	Not Completed	0%
28.	Search and Filter functionality	Not Completed	0%
29.	Sprint Testing	Not Completed	0%

Sprint 4

30.	Recommendation System Features	Not Completed	0%
31.	Customization option Features	Not Completed	0%
32.	Admin Side Orders Listing of Each user	Not Completed	0%
33.	Testing All the Features	Not Completed	0%

34.	Document the Entire Development	Not Completed	0%
-----	---------------------------------	---------------	----

35.	System Testing	Not Completed	0%
36.	Review Documentation	Not Completed	0%
37.	Review the Project	Not Completed	0%
38.	Submit the Project	Not Completed	0%

Table 11: Progress Table

Go back to the Progress review section [Click here.](#)

7.8 Technology Used

7.8.1 UI Framework React JS

In this project, React JS is used as a UI framework for the applications. It is known for its flexibility in building interactive user interfaces. It allows us to create reusable UI components, making the development process more manageable. React is a component-based architecture that enables us to build complex user interfaces with less code, leading to a more maintainable and scalable application (simplilearn, 2023).

7.8.2 Programming Language JavaScript

In this project, I have chosen JavaScript as our programming language. JavaScript is flexible and powerful, and it's commonly used for making web pages interactive. It's great for creating web apps because it's light and easy to run. JavaScript lets us add dynamic features to our web pages, like animations, interactive forms, and advanced user interfaces, without needing to refresh the page. (developer.mozilla.org, 2023).

7.8.3 Backend framework Node JS

In this project, I have implemented Node.js as the framework for backend development. Node.js is a strong and efficient platform for server-side programming, built using Chrome's JavaScript engine. It's great for building scalable network applications in JavaScript, fitting both front-end and back-end needs. Node.js works with an event-driven model that doesn't block input/output operations, making it light and fast. (nodejs.org, 2023).

7.8.4 APIs

c) Weather API

A Weather API gives us live weather details for where the user is. It tells us things like the temperature, if it's raining, sunny, or snowing, and what the weather will be like later. With this info, you can suggest food that fits the weather. For instance, on cold days, you could recommend warm soups or drinks. so This API is used in this project (visualcrossing, 2023).

d) Map API

In this project, I have implemented map API for finding current customer locations. Application lets customers pick their location right on the interface. A Map API, like Google Maps API, helps put maps into web. It can show maps, mark places, and give directions. Using a Map API makes it easier for customers to spot and confirm where they are. This is important for giving them accurate and quick service (pubnub, 2023).

7.8.5 Database MongoDB

In this project I have chosen MongoDB as database. MongoDB is a popular NoSQL database known for its flexibility and scalability. Unlike traditional relational databases, MongoDB stores data in flexible, JSON-like documents, which makes data integration for certain types of applications faster and easier. this document model is particularly effective for handling large volumes of data and complex data structures, which are common in modern web applications (Gillis, 2023).

Go back to the technology used section [Click here](#)

7.9 Frontend File and Folder Structure

```

import React from "react";
import { Route, Routes } from "react-router-dom";
import HomePage from "./pages/Home/HomePage";
import MenuPage from "./pages/Menu/MenuPage";
import FoodPage from "./pages/Food/foodPage";
import CartPage from "./pages/Cart/CartPage";
import LoginPage from "./pages/Login/LoginPage";
import AdminDashboard from "./Admin/AdminDashboard";
import RestaurantPage from "./Admin/Pages/RestaurantPage/RestaurantPage";

export default function AppRoutes() {
  return (
    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/menu/:restaurantId" element={<MenuPage />} />
      <Route path="/cart" element={<CartPage />} />
      <Route path="/login" element={<LoginPage />} />
      <Route path="/admin/*" element={<AdminDashboard />} />
      <Route path="/restaurants" element={<RestaurantPage />} />
      <Route path="/foods" element={<FoodPage />} />
    </Routes>
  );
}

```

Figure 47: frontend file and folder structure

7.10 Backend File and Folder Structure

```

import dotenv from "dotenv";
dotenv.config();
import express from "express";
import cors from "cors";
import foodRouter from "./routers/food.router.js";
import userRouter from "./routers/user.router.js";

import { dbconnect } from "./config/database.config.js";
dbconnect();

const app = express(); //create an Express application
app.use(express.json()); //telling the express app to use json
app.use(
  cors({
    credentials: true, //server allows credentials like authentication info
    origin: ["http://localhost:3000"],
  })
);

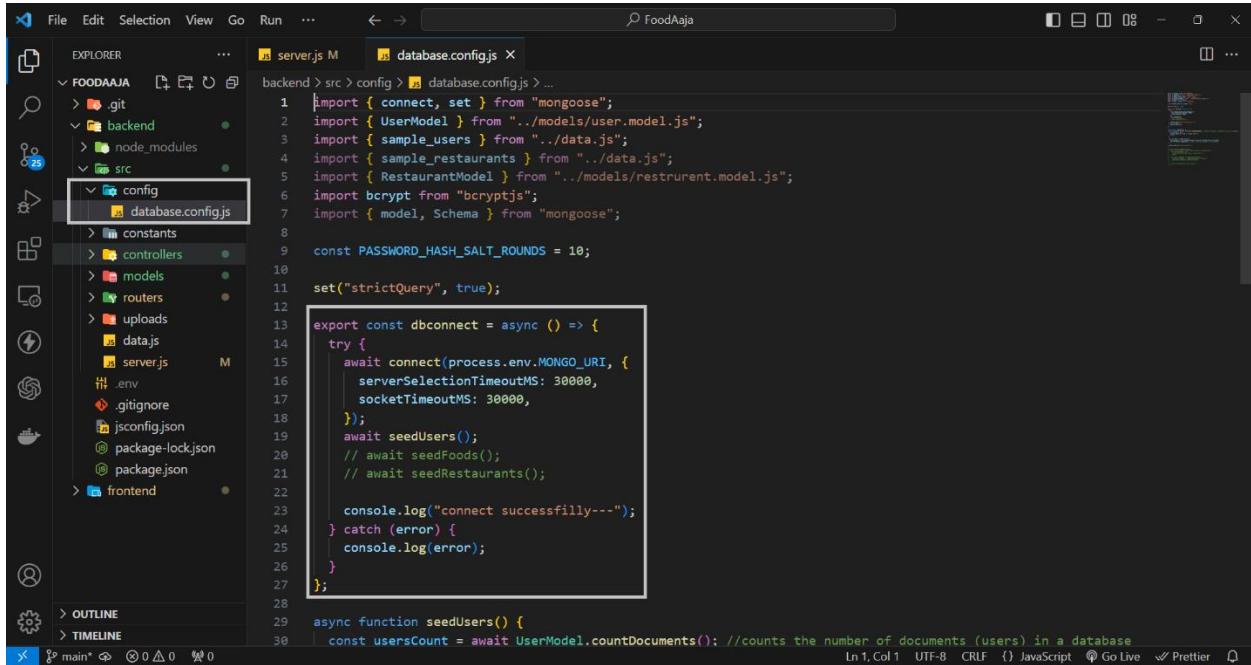
app.use(express.json({ limit: "50mb" })); //can adjust the payload size limit by configuring the body-parser middleware
app.use("/api/restaurants", foodRouter);
app.use("/api/users", userRouter);

const PORT = 5000;
app.listen(PORT, () => {
  console.log("listening on port " + PORT);
});

```

Figure 48: backend file and folder structure

7.11 Database connection



```

1 import { connect, set } from "mongoose";
2 import { UserModel } from "../models/user.model.js";
3 import { sample_users } from "../data.js";
4 import { sample_restaurants } from "../data.js";
5 import { RestaurantModel } from "../models/restaurant.model.js";
6 import bcrypt from "bcryptjs";
7 import { model, Schema } from "mongoose";
8
9 const PASSWORD_HASH_SALT_ROUNDS = 10;
10
11 set("strictQuery", true);
12
13 export const dbconnect = async () => {
14   try {
15     await connect(process.env.MONGO_URI, {
16       serverSelectionTimeoutMS: 30000,
17       socketTimeoutMS: 30000,
18     });
19     await seedUsers();
20     // await seedFoods();
21     // await seedRestaurants();
22
23     console.log("connect successfully---");
24   } catch (error) {
25     console.log(error);
26   }
27 };
28
29 async function seedUsers() {
30   const usersCount = await UserModel.countDocuments(); //counts the number of documents (users) in a database

```

Figure 49: Database connection

7.12 Admin Folder and Models

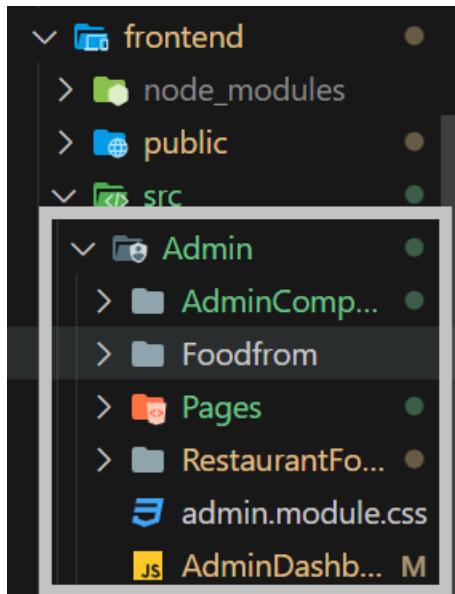


Figure 50: Frontend Admin folder

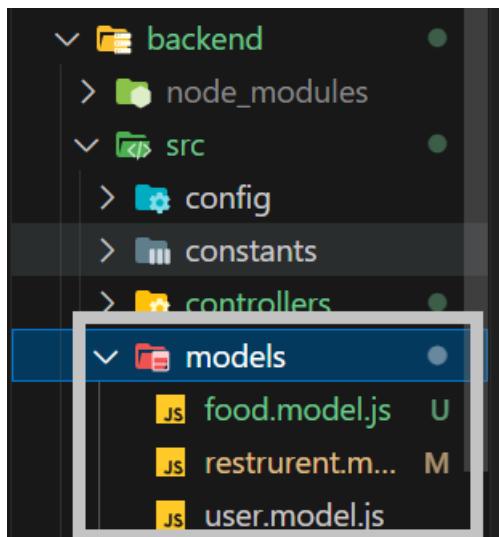


Figure 51: backend models