Major Project Report on

Web Based Restaurant Search Application Using User Location

Achalla Sampath (201900022)

In partial fulfillment of requirements for the award of degree in Bachelor ofTechnology in Computer Science and Engineering
(2023)

Under Project Guidance of
Internal Reviewer:
Mrs. Chitrapriya Ningthoujam
Assistant Professor (SG)
Department of Computer Science and Engineering
External Guide:
Mr. Sathish Kumar
(Project Manager, Data Lake Solutions)
&
Mr. Ashok
(Team Lead,Data Lake Solutions)

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY
(A constituent college of Sikkim Manipal University)
MAJITAR, RANGPO, EAST SIKKIM – 737136

PROJECT REVIEW CERTIFICATE

This is to certify that the work recorded in this project report entitled "**Developing Web-Based Restaurant Search Application Using GPS (Cuisine Compass)**" has been carried out by Mr. Achalla Sampath (201900022) of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment ofthe requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Major Project Viva Examination.

Mrs. Chitrapriya Ningthoujam
Assistant Professor(SG)
Department of Electrical & Electronics Engineering
Sikkim Manipal Institute of Technology
Majitar,
East Sikkim – 737136.

CERTIFICATE OF ACCEPTANCE

This is to certify that the below mentioned student of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) has worked under the supervision of Mr. Sathish Kumar of Data Lake Solutions from 27/02/2023 to 16/06/2023 on the project entitled "**Developing Web-Based Restaurant Search Application Using GPS(Cuisine Compass)**"

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Project Venue
201900022	Achalla Sampath	Datalake Solutions 2nd Floor, JQ Chambers Jayabheri Enclave, Gachibowli Hyderabad, Telangana 500032

DECLARATION

We the undersigned, hereby declare that the work recorded in this project report entitled **"Developing Web-Based Restaurant Search Application Using GPS(Cuisine Compass)"** in partial fulfillment for the requirements of award of B.Tech in Computer Science & Engineering from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at Kolkata under the supervision and guidance of **Mr. Sathish Kumar and Mr. Ashok** from **Data Lake Solutions.** The results of this investigation reported in this project have so far not been reported for any other Degree / Diploma or any other Technical forum. The assistance and help received during the courseof the investigation have been duly acknowledged.

Achalla Sampath (Reg. No.-201900022)

ACKNOWLEDGEMENT

We would like to thank **Mr. Sathish Kumar and Mr. Ashok** from **the Data Lake Solutions** for his guidance during the development ofour project. His support was invaluable to the project's success, and I look forward to applyingthe skills I have learned under his mentorship.

We take this opportunity to acknowledge indebtedness and deep sense of gratitude to my guide **Mrs. Chitrapriya Ningthoujam** whose valuable guidance and kind supervision gave us throughout the course which shaped the present work as it shows.

I/We pay our deep sense of gratitude to **Prof. (Dr.) Udit Kr Chakraborty, Professor & H.O.D, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and provided all support required.

We are obliged to our project coordinators **Mrs. Chitrapriya Ningthoujam, Mr. Sourav Paul and Mr. Santanu Kumar Misra** for elevating, inspiration and supervising in completion of our project.

We would also like to thank any other staff of **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support andguidance that has helped us in completion of our project.

Achalla Sampath (Reg. No.-201900022)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Major Project/External /B.Tech/SA77/2023
2	Title of the Report	Restaurant Search Application
3	Type of Report	Technical
4	Author	Achalla Sampath
5	Organizing Unit	Data Lake Solutions
6	Language of the Document	English
7	Abstract	This project aims to create a web-based application to search restaurants based on user input and location. Users can increase search efficiency by applying distance, rating, and cuisine filters.
8	Security Classification	General
9	Distribution Statement	General

LIST OF CONTENTS

Chapter	Title	Page No.
	Abstract	
1	**Introduction**	
	1.1 General Overview of the Problem	1-1
	1.2 Literature Survey	2-3
	1.3 Problem Statement	4-5
	1.4 Software Requirements Specifications	6-12
	1.5 Proposed Solution Strategy	13-14
	1.6 User's Manual	14-14
	1.7 Organization of report	15-15
2	**Design Strategy for the solution**	
	2.1 Use Case Diagram	16-17
	2.2 Class Diagram	18-18
3	**Implementation Details**	
	3.1 Technology Stack	19-19
	3.2 Pseudo Code	19-21
	3.3 REST API Routes	22-25
4	**Results and Discussions**	26-36
5	**Testing and Validation**	
	5.1 Frontend Validations	37-44
	5.2 Backend Unit Testing	45-45
6	**Summary and Conclusion**	
	6.1 Summary of achievements	46-46
	6.2 Main difficulties encountered	47-47
	6.3 Limitations of the project	48-48
	6.4 Future Scope of work	49-49
	6.5 Conclusion	50-50
7	**References**	51-51

List of Figures

Fig. No.	Figure Name	Page No.
2.1.1	Use Case Diagram with an actor admin	16 – 16
2.1.2	Use Case Diagram with an actor user	17 – 17
2.2.1	Class Diagram	18 – 18
4.1.1	Default Page	26 – 26
4.1.2	Search Result Page	27 – 27
4.1.3	Search Result when user input is restaurant name	27 – 27
4.1.4	Search Result when user input is dish name	28 – 28
4.1.5	Search Result when user input is dish name and distance	28 – 28
4.1.6	Details of particular restaurant	29 – 29
4.1.7	Reviews of particular restaurant	29 – 29
4.1.8	Sample Review Form	30 – 30
4.1.9	Message after submitting the review	30 – 30
4.2.1	Admin Login Page	31 – 31
4.2.2	Restaurant Registration	31 – 31
4.2.3	Admin Restaurant Dashboard	32 – 32
4.2.4	Searching Particular Restaurant	32 – 32
4.2.5	Message displayed when admin sets store to offline	33 – 33
4.2.6	Message displayed when admin sets store to online	33 – 33
4.2.7	Admin side restaurant details	34 – 34
4.2.8	Message displayed when admin edited the fields	34 – 34
4.2.9	Admin side review section	35 – 35
4.2.10	Message displayed when admin approved review	35 – 35
4.2.11	List of admins	36 – 36

LIST OF TABLES

Table. No.	Table Name	Page No.
1.2.1	Literature Survey	2 – 3
3.1.1	Technology Stack	19 – 19
3.3.2	REST API routes of Customer	22 – 23
3.3.3	REST API routes of Merchant	24 – 25
5.1.1	Test Cases of Search functionality	37 – 39
5.1.2	Test Cases of Login functionality	40 – 40
5.1.3	Test Cases of Register Restaurant functionality	41 – 42
5.1.4	Test Cases of Update Restaurant functionality	43 – 43
5.1.5	Test Cases of Add Review functionality	44 – 44

ABSTRACT

The Indian restaurant industry is a thriving sector of the food and beverage industry. Since India is densely populated, with 1.4 billion, it significantly impacts the restaurant industry. Indian cuisine is popular among Indians and non-Indians alike, and the demand for Indian food has grown as the Indian population has grown. As a result, many restaurants are evolving across India every day, increasing employment and the food sector's revenue. As a result, people know less about restaurants and need clarification about selecting restaurants, resulting in the wrong choices.

This project aims to create a web-based application to search restaurants based on user input and location. Users can increase search efficiency by applying distance, rating, and cuisine filters. Along with the search functionality, users can add ratings and reviews to the restaurant so that it will be helpful for other users. On the other hand, Admin can add new restaurants, modify the existing ones, and able to verify the user reviews.

1. INTRODUCTION

1.1 General Overview of the Problem

The Indian restaurant industry is a significant part of the global food service. It encompasses a wide range of restaurants, from small mom-and-pop shops to large chain restaurants that serve a variety of Indian cuisine. The industry has grown significantly in recent years due to globalization, immigration, and a growing interest in ethnic cuisine. In India, the restaurant industry is one of the economy's fastest-growing sectors, with a projected annual growth rate of 10%. This growth is driven by a growing middle class, increasing disposable incomes, and a desire for more diverse dining options. The industry is also becoming more competitive, with new players entering the market and existing players expanding their offerings. Overall, the Indian restaurant industry is poised for continued growth in the coming years—this growth and competition cause trouble for familiar people in selecting the perfect restaurant based on their preferences.

Every day many restaurants are opened across different areas of Indian cities, and every area consists of multiple restaurants which offer a wide variety of cuisines. This number will increase in major cities like Mumbai, Hyderabad, and Delhi. People who are residents or well-known about the place can choose the best restaurant based on their preference. But a newcomer will need help in selecting a restaurant. By the suggestion of others or attracted by offers and ambiance, they need to pay for the food they dislike entirely.

To resolve this problem, we need an application where users can search restaurants based on input entered in the search bar by the user. The input may include the restaurant name (or), cuisine type (or), distance, and rating. The application should process the user input and should produce the desired output. The application should also allow users to apply filters like cuisine, rating, and distance which helps to increase search efficiency and helps people select the restaurants for casual weekend dinners or special events based on their requirements and taste. And the application also allows users to provide ratings and reviews for the available restaurants so that it can be helpful for other users.

1.2 Literature Survey

Sl.No.	Author	Paper and Publication Details	Findings	Relevance of Project
1.	Aureliano-Silva, Leonardo & Leung, Xi & Spers, Eduardo Eugênio.	The effect of online reviews on restaurant visit intentions: applying signaling and involvement theories. Journal of Hospitality and Tourism Technology. ahead-of-print. 10.1108/JHTT-06-2020-0143 Date: August,2021	Recognize that ratings and reviews influence user restaurant selection and that restaurants with better ratings and positive reviews attract more other people	Reviews are very important when choosing a restaurant. As a result, a more extensive and generic review mechanism is required.
2.	Kim, Jaewook & Lee, Minwoo & Kwon, Wooseok & Park, Hyekyung & Back, Ki-Joon.	Why am I satisfied? See my reviews – Price and location matter in the restaurant industry. International Journal of HospitalityManagement.101. 103111. 10.1016/j.ijhm.2021.103111. Date: February,2022	Explains why modestly priced restaurants and restaurants in well-known and busy places attract a greater number of customers	Restaurants in the database should be reasonably priced, and their location should be easily accessible to the user

Sl.No.	Author	Paper and Publication Details	Findings	Relevance of Project
3.	1. Isabela, Erika & Drona, Jennifer & Fadhilah, Nailatul & Tanoto, Dian & Harefa, Jeklin & Prajena, Gredion & Chowanda, Andry & Alexander	NYAM: An Android Based Application for Food Finding Using GPS. Procedia Computer Science. 135. 393-399. 10.1016/j.procs.2018.08.189. Date: August,2018	How to make the application user-friendly and how to use GPS for calculating the accurate position of the user and identifying the restaurants nearby.	The crucial functionality of the application is used to get user latitude and longitude and calculate the distance between the user and restaurants using the haversine formula.
4.				

Table 1.2.1 Literature Survey

1.3 Problem Definition

Major cities such as Mumbai, Delhi, Hyderabad, and Bangalore are heavily inhabited, and heavy traffic makes it difficult to travel from one location to another. Whenever people decide to eat outside, they try to order food online or visit a nearby restaurant. Some prefer to order food online, while others prefer takeaway from nearby restaurants. Everything will be good until the person is local or familiar with the restaurant. It is challenging for a newcomer to the area to choose a restaurant nearby to order food or eat in the restaurant.

Usually, people new to the area rely on word-of-mouth recommendations from friends or online. Still, friend recommendations are entirely personal and vary from one person to another. Sometimes they will be ended up in a bad situation if their food taste utterly different from another friend who suggested it. On the other hand, Search engines often return a long list of results. The extensive lists of restaurants, menus, and reviews can lead to decision fatigue, making it challenging to make a confident choice. Users may need help to narrow down their options and make a final decision. which can be overwhelming and time-consuming. Since there are so many restaurants, it can take time to find the right one.The online blogs recommended by the search engines suggest some restaurants, but it contains limited information about them. You may only get basic details like the name, location, and contact number. Sometimes information mentioned in the blogs can be incomplete, inaccurate, or outdated. Users might encounter incorrect operating hours, closed restaurants still listed, or bygone menus, leading to frustration and wasted time.

While many restaurant search applications have extensive databases, there can still be limitations in terms of coverage. They might only include some restaurants in a specific area or may focus primarily on popular locations, leaving out smaller or lesser-known establishments. While restaurant search applications often provide filtering options, finding specific requirements can still be challenging. When a user searches for a restaurant, they consider various factors. So they may use multiple search applications. In contrast, restaurant search applications often provide filtering options such as price range, distance, cuisine types, and so on; finding specific requirements can still be challenging. For example, searching for restaurants with specific dietary options or specialized cuisines may yield incomplete or inaccurate results. The application may need a comprehensive database or search filters to cater to all user preferences. Unfortunately, current search applications have minimal filters, and users cannot select numerous filters at the exact moment.

Users who attempt to buy food online or visit a restaurant rely on third-party food delivery apps or search engines such as Google. Users may be able to choose based on user reviews and ratings of restaurants mentioned in the application. There are some instances where the reviews can also be unreliable. Some users may leave biased or inaccurate reviews or be manipulated by competitors leaving negative reviews or businesses posting fake positive reviews. This can mislead users and make it difficult to trust the authenticity of the reviews, making it difficult for other users to make decisions.

1.4 Software Requirements Specifications

1.4.1 Functional Requirements:

1.4.1.1 Login:

- **Inputs**: Username, Password
- **Outputs**: Invalid Username/Password
- **Description**: Admin can log into the cuisine compass administration dashboard by entering their username and password on the login form. From the dashboard, the administrator can monitor many tasks, such as adding new restaurants, modifying the existing ones, and verifying the reviews the users submit.

1.4.1.2 Add Restaurant:

- **Inputs**: Restaurant Name, Categories, Cuisines, Minimum Price, Maximum Price, Open Time, Close Time, Address, Latitude, Longitude, Pincode
- **Outputs:** Successfully Registered / Store is Present / Invalid Input
- **Description:** In this application, only the administrator can add new restaurants by providing information in the required registration form fields, and the form includes validation for each field. The application will examine each field and generate responses accordingly, and an alert will be shown if the provided information needs to be corrected.

1.4.1.3 Update Restaurant:

- **Inputs**: Open Time, Close Time, Minimum Price, Maximum Price, Cuisines, Categories, Services / Address, Latitude, Longitude, Pincode.
- **Outputs:** Updated Successfully
- **Description:** This application update restaurant capability allows the admin to modify existing restaurant details. The admin can update restaurant details like cuisine, categories, timing, pricing, address, Pincode, longitude, and latitude. Every field on the form has to be validated to guarantee that only legitimate inputs are accepted**.**

1.4.1.4 Reviews Action:

- **Inputs:** No Input
- **Outputs:** Review Approved / Review Rejected
- **Description:** This application reviews action function allows the administrator to verify the user reviews manually and take actions accordingly. Whenever a user enters a review, the review status will be registered as a new state in the database, and this review will be invisible to other users. It will be visible if it states changes to the accepted state.

1.4.1.5 Admin Store Search:

- **Inputs:** Store Name
- **Output:** Matched Stored
- **Description:** In the administrator dashboard, the store search function helps the administrator to search for a particular restaurant out of many restaurants available in the application database, and it also helps the administrator to select a restaurant for performing actions to invoke necessary functions.

1.4.1.6 Store Status:

- **Inputs**: No Inputs
- **Outputs:** Store Online / Store Offline
- **Description:** This application store status function allows the administrator to change the Restaurant's status. If the Restaurant's status is online, it will be displayed during the user search operation. If the status of the Restaurant is mentioned as offline, it will not be displayed during the user search operation. So that users will get information about the Restaurant which are currently operating.

1.4.1.7 Restaurant search by store name:

- **Inputs**: Restaurant Name
- **Outputs:** Matched Restaurant
- **Description:** The cuisine compass application users can use the search feature to look up restaurants by name. The web page retrieves the Restaurant that corresponds to the user's input. The user can perform necessary functions like gathering information about the Restaurant and reviewing the Restaurant.

1.4.1.8 Restaurant search by dishes / cuisines:

- **Inputs**: Dishes / Cuisines
- **Outputs:** Matched Restaurants.
- **Description:** Users can utilize the search function of the cuisine compass application. Whenever a user mentions any particular cuisine like Chinese or Italian, or Indian or dishes like biryani or pongal on the search input bar, the algorithm of the search application can display the restaurants in a nearby location that satisfy the user's requirements.

1.4.1.9 Restaurant search by distance / rating / pricing:

- **Inputs**: Rating / Location / Price
- **Outputs:** Matched Restaurants
- **Description:** Users of the cuisine compass application can do Restaurant searches by considering restaurant location, rating, or pricing. Simply users need to enter either price range or nearby location or ratings. The search algorithm of the application can fetch restaurants based on user input.

1.4.1.10 Increasing search efficiency by applying filters:

- **Inputs**: No Input
- **Outputs:** Matched Restaurants
- **Description:** This application provides users with filters, which helps in increasing the search efficiency of restaurant searches. Various filters like pricing, rating, cuisine, categories, services, and distance can be applied individually, or they can be applied at a time to the search function.

1.4.1.11 Show Restaurant Details:

- **Inputs**: No Input

- **Outputs:** Name, Timing, Address, Cuisine, Rating and Reviews
- **Description:** The cuisine compass application users can retrieve information about various restaurants from the application's database. Whenever the user selects a particular restaurant the webpage displays data related to one Restaurant. Mainly the data consists of restaurant timing, price range, services available, cuisine served in the Restaurant, restaurant address, and some of the images of the Restaurant.

1.4.1.12 Add Review:

- **Inputs**: Email id, Name, Review

- **Outputs:** Review Submitted Successfully / Incomplete Payload

- **Description:** This application allows users to share their experiences about the restaurants they visited with other users by providing ratings and reviews. Users can provide ratings and reviews to the particular Restaurant by entering their name, email id, and review in the corresponding fields. The user review will be stored in the database only after verifying every user review field, and this review will be displayed to other users only after changing to approved status, which the application administrator can do.

1.4.2 Non Functional Requirements:

1.4.2.1 Performance Requirements:

- The System must be capable of handling significant number of books and users without fail
- Responses to generating insights must appear on the screen in no more than 3 seconds.
- It must manage expected and unexpected errors in ways that avoid data loss and prolonged unavailability. As a result, it should include error control to detect fraudulent usernames or passwords.

1.4.2.2 Security Requirements:

- The System's database will be protected by AWS security, and AuthGuard will secure routes.
- Regular users must only view information and cannot edit or alter it.
- The System will have many different kinds of users, each with their own set of access restrictions.
- Adequate user authentication is required.
- There should be distinct accounts for admin and members so that no member can access the databaseand only the admin can modify it.

1.4.2.3 Software Requirements:

- Database: Aws dynamo DB
- Language: Python
- Frontend Development Tools: HTML, CSS, Bootstrap
- Frontend Frameworks: AngularJs
- Backend Frameworks: AWS Lambda and AWS API Gateway
- Development Environment: Visual Studio Code
- Version Control: GitHub

1.4.2.4 Hardware Requirements:

- RAM: 256 MB or more
- Hard Disk: 1 GB or more

1.5 Proposed Solution Strategy

To address the limitations of traditional restaurant search methods, such as word-of-mouth recommendations, extensive lists of restaurants that cause decision fatigue, limited information from online blogs, incorrect and outdated information from webpages, limited filtering options, and manipulated reviews, a solution strategy can be developed using Python, Angular Js, boto3 and AWS(Amazon Web Services).

1. Create the Database Schema:

Creating the database schema is the first step in establishing an application, which involves determining the many entities stored within the database, including the restaurant names, addresses, cuisines, services, user reviews, ratings, latitude, and longitude. Following by the identification of the entities and outlining the relationships between them.

2. Establishing the database:

Setting up the database is the next step after designing the schema. We use DynamoDB and S3 storage for the database, popular services Amazon Web Services(AWS) run. Amazon DynamoDB is a fully managed, serverless, key-value NoSQL database designed to run high-performance applications at any scale. It offers built-in security, continuous backups, and data import and export tools. Amazon S3 is an object storage service that is used for storing images related to the restaurant

3. Create an AWS Lambda Function:

After configuring the database, a lambda function should be created to establish a connection between front-end and back-end applications. AWS Lambda is a serverless computing service provided by AWS. It allows developers to run code without having to provision or manage servers. An Aws lambda function processes user data through API calls, gets the required data from dynamoDB and s3 services, and sends the processed data as output through API calls to the front-end application.

4. Create APIs:

After configuring the database and AWS Lambda functions, configuring APIs is the following process. RESTful APIs, which enable real-time two-way communication applications, are suitable for this application and can be developed using AWS API Gateway. Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

5. Create User Interface:

AngularJS is a robust JavaScript framework that enables dynamic and responsive web application development. Businesses can utilize AngularJS to create a user-friendly interface where users can search for restaurants by using restaurant names (or) cuisine (or) dishes (or) locations and able to apply various filters, which increases search efficiency and also able to provide ratings and reviews to the restaurants. Similarly, admins can sign in to their accounts and access the administrator dashboard to monitor all the application functions.

1.6 User's Manual

Users of the cuisine compass application can perform various functions, including restaurant searching, getting details of restaurants, and Providing ratings and reviews to the restaurant they visited recently.

The user does not need to set up an account; they can enter the input in the application's search bar. Users can mention any details related to the restaurants, like the restaurant name (or) location (or) dishes and cuisine. The search algorithm will fetch suitable restaurants in nearby locations.

Users can get more restaurant information by clicking on the restaurant box. Users will be redirected to another webpage that displays all the information related to the restaurant, like name, images, address, the cuisine offered in the restaurant, opening and closing time, current rating of the restaurant, and reviews that other users gave. Users can also add ratings and reviews to the restaurant by clicking the add review button. A User can enter their name, email ID, and review on the restaurant. The review will only be visible to the other users after the administrator verifies it.

Admin of the application with his unique username and password had full access to the application. Once the admin has checked in, they can add new restaurants to the application database by providing restaurant details in respective fields. With the admin dashboard, the admin can monitor all the user reviews and act accordingly, and only the reviews accepted by the admin are visible to general users. In the restaurant monitor page admin able to make changes to the various restaurant fields, and the admin can set the restaurant offline so that it cannot be visible to the users during their search process.

1.7 Organization of report

- Chapter 1 Introduction contains some sub-sections. They are a General Overview of the Problem, Literature Survey, Problem Definition, Software requirements and specifications, and Proposed Solution Strategy.
- Chapter 1.1 General Overview of the Problem Provides an overview of the problem related to restaurant search applications and the limitations of traditional methods for searching for restaurants.
- Chapter 1.2 Literature survey discusses the research papers, details of author & paper publication, and relevance of the project.
- Chapter 1.3 Problem definition discusses the problems which general users face.
- Chapter 1.4 Software requirements and specifications outline the necessary functions and features of the system, including both functional and non-functional requirements.
- Chapter 1.5, Proposed Solution Strategy, discusses the solution to the problems by following some strategies.
- Chapter 1.6 User manual provides instructions and guidelines for operating a software website project.
- Chapter 2 this section pertains to the design approach adopted for the project. It encompasses the high-level architecture of the system, including the class diagram, and uses a case diagram of the admin and user functionalities.
- Chapter 3, the implementation details refer to the technology stack employed in the project's development, the API routes, and the frontend web pages refer to the actual visual interface presented to the user.
- Chapter 4, Results and Discussions, outlines the frontend webpages of the cuisine compass application.
- Chapter 5, Testing and, Validations, gives an outline of validations used and backend service layer testing to check that the data is processed correctly and that the system can handle various scenarios.

2. DESIGN STRATEGY FOR THE SOLUTION

2.1 Use Case Diagram

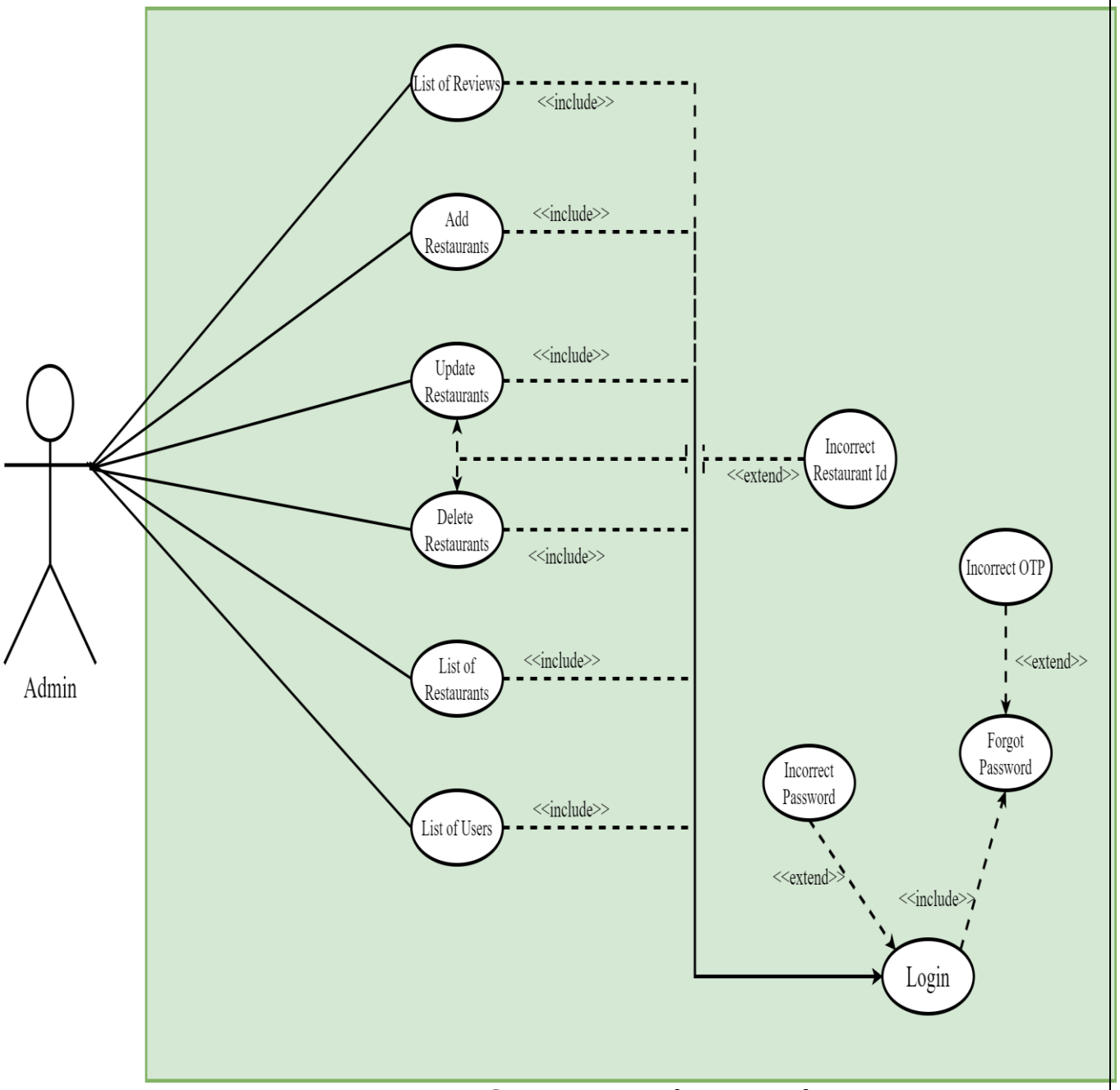


Fig. 2.1.1 Use Case Diagram with an actor admin

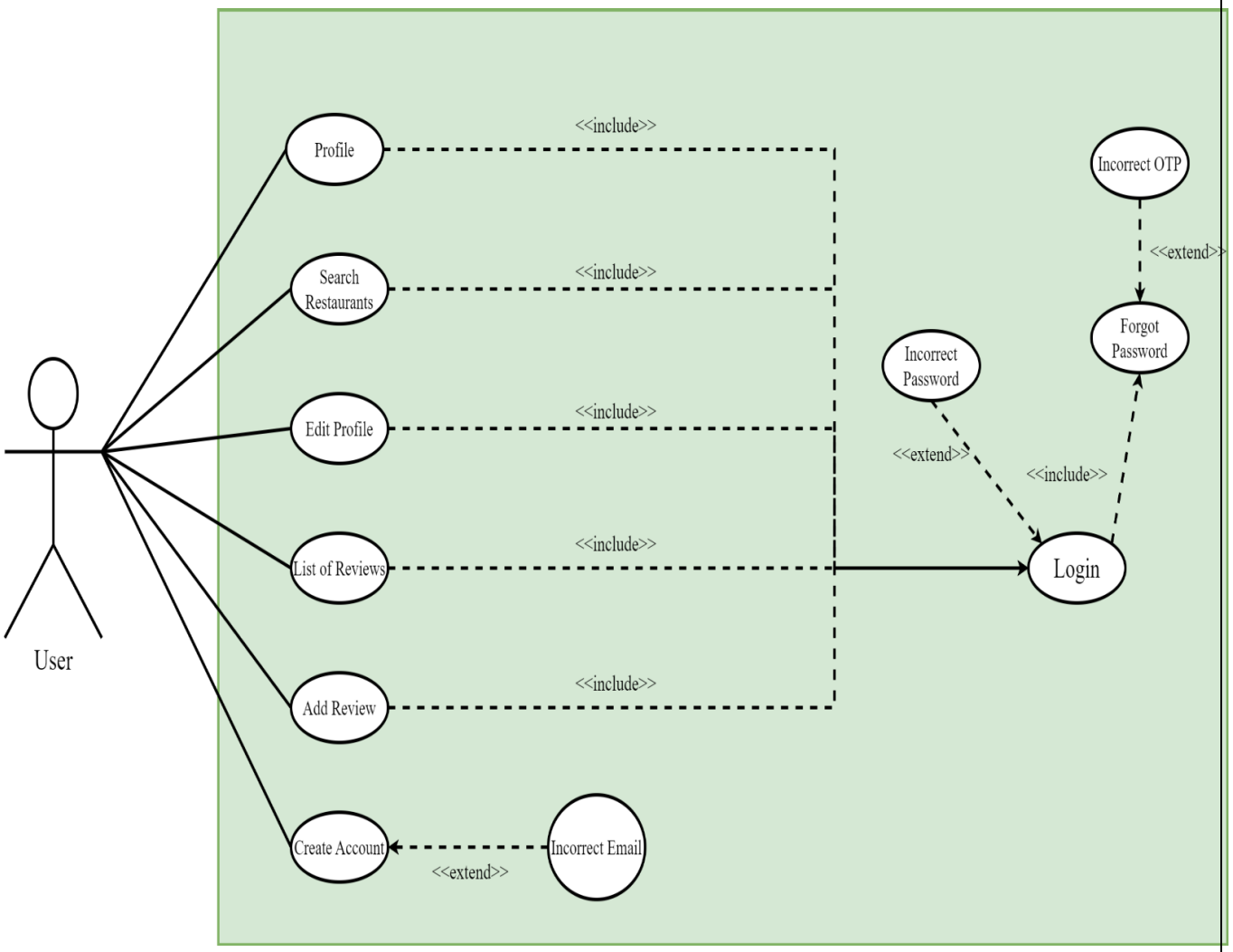


Fig. 2.1.2 Use Case Diagram with an actor user

2.2 Class Diagram

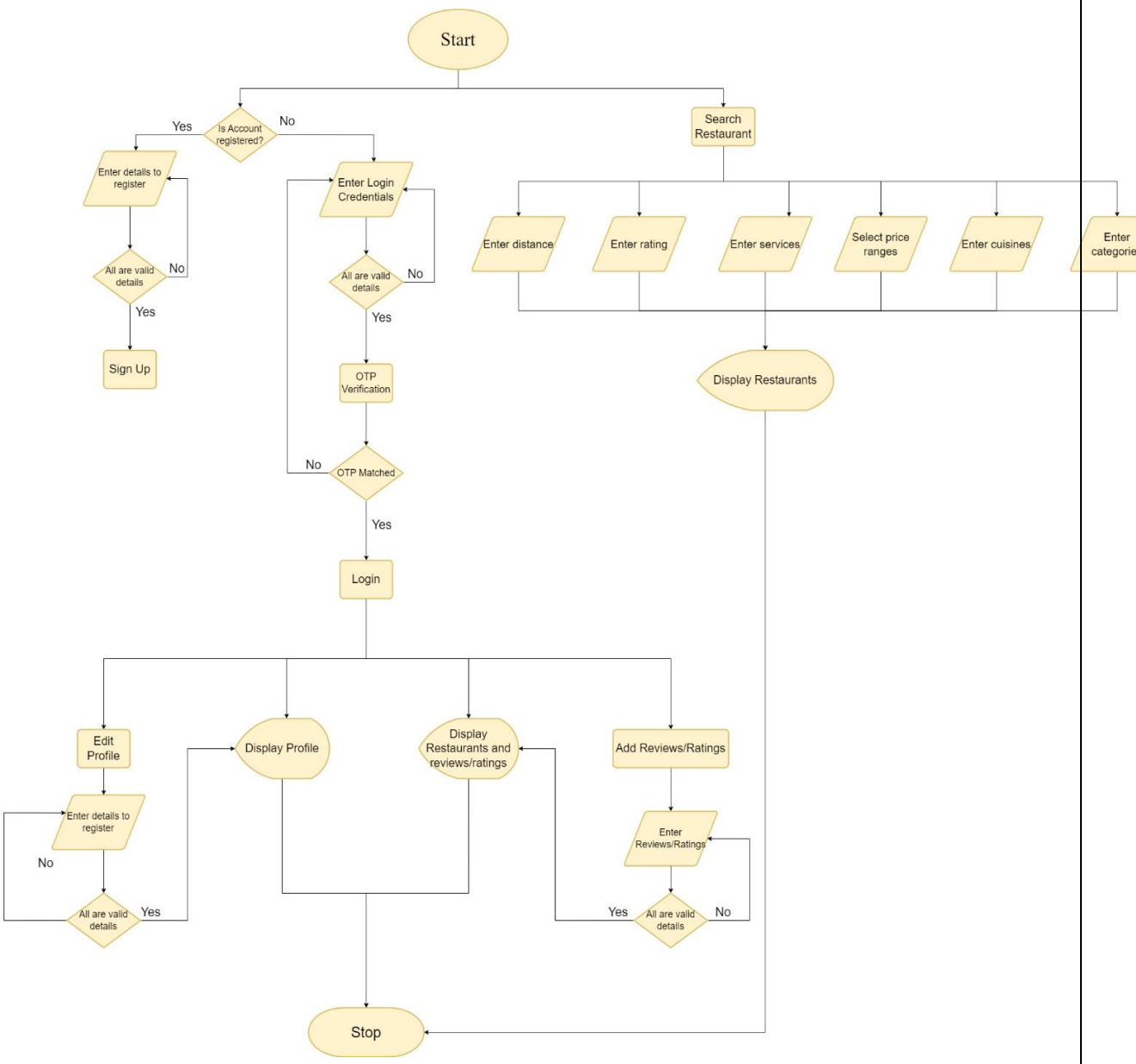


Fig. 2.2.1 Class Diagram

3. IMPLEMENTATION DETAILS

3.1 Technology Stack

Division	Technology Stacks
Front End	HTML, CSS, Bootstrap, JavaScript ,
Back End	Amazon Web Services
Database	AWS dynamo DB and S3 Services
Testing	Manual Testing
Development Environment (IDE)	Visual Studio Code

Table 3.1.1 Technology Stack

3.2 Pseudo Code

3.2.1 Pseudo code for Search functionality

```
FUNCTION Search():
        SortByDistance()
    IF userOperation==1 Then:
        NormalSearchFunction()
     END IF
     IF UserOperation==0 Then:
        FilterSearch()
     END IF
END FUNCTION
```

```
FUNCTION SortByDistance():

        Count= number  of restaurants in database
        IF count>0 then
                merchantList=restaurants of database
        END IF
        FOR I in range of count:
                Cal_dist=distance_calculation(user_latitude,user_longitude,
                store1_latitude,store1_longitude)
                merchantDictionary[cal_dist]=MerchantList[i]
        END FOR
        FOR I in range of length of merchant dictionary:
                sortedList[i]=merchantDictionary[i]
        END FOR
           Return sortedList

END FUNCTION

FUNCTION DistanceCalculation:
        dlon=(store_longitude)-(user_longitude)
        dlat=(store_latitude)-(user_latitude)
        a=sin(dlat/2)**2 + cos(user_latitude)*cos(store_latitude)*sin(dlon/2)**2
        c=2*asin(sqrt(a))
        r=6371
        return(c*r)
END FUNCTION

FUNCTION NormalSearchFunction():
        FOR I in range of length of merchantSortedList:
                IF userInput==merchantSortedList[Name] :
                        Return merchantSortedList[i]
                END IF
                ELSE:
                        InputFilterExtraction
END FUNCTION
```

```
FUNCTION InputFilterExtraction():
        IF userInput in FilterDictionary Then:
                FilterFlag=1
        END IF
        IF UserInput in MerchantDictionary Then:
                MerchantFlag=1
        END IF
        IF FilterFlag==1 Then:
                SubFilterFun()
        END IF
END FUNCTION

FUNCTION SubFilter():
        FOR I in range of length of input:
                IF input[i] in Cuisines Then:
                        CuisineFlag=1
                END IF
                IF input[i] in Categories Then:
                        CategoriesFlag=1
                END IF
                IF input[i] in Distance Then:
                        DistanceFlag=1
                END IF
          END FOR
        Return FlagsList
END FUNCTION
```

3.3 REST API Routes

The Cuisine Compass application API's make it possible for general users to automate the traditional search, making it a simple and efficient way for searching. Admin can able to add, update and deactivate the restaurants using these API's. Users can also use the different search scenarios for searching the restaurants and also able to add review to the restaurants and applying filters. These API's are designed with AWS API Gateway and the routes are easy to integrate into existing systems because they are RestFul and follow standard HTTPs guidelines.

3.3.1 Customer

In this application the customer API offers endpoints for Search, Apply, View and Add actions on instances. With the use of this API users may simply search restaurants, Apply filters on it, View their details and able to add reviews to them.

Routes	Method Type	Request Body	Description
customer/ view-merchant	POST	{ "op": "1", "M_ID": "M-20230419101828", "user_id": "sampath3927@gmail.com", "user_name": "sampath", "user_rating": "5", "user_review": "hotel is neat and food is awesome and staff is professional and price is nominal" }	This endpoint allows the customer to view details of the restaurant as well as allow customer to add reviews to the restaurant.

/customer/search	POST	{
		"op": "1",
		"user_inp": "restaurants near me",
		"user_latitude": "17.4563154",
		"user_longitude": "78.37199",
		"m_distance": 0,
		"m_rating": 0,
		"m_price": 300,
		"m_category": "",
		"m_cusine": "",
		"m_service": ""
		}
			This endpoint returns the list of restaurants according to the the user input criteria.

3.3.2 Merchant

In this application the merchant API offers endpoints for actions such as adding restaurant details, updating restaurant details, activating or deactivating the restaurants, Accepting or rejecting user reviews.

Routes	Method Type	Request Body	Description

/merchant/merchants-view	POST	{
		"op": "2",
		"M_ID": "M-20230419101828",
		"search_inp": "chutneys"
		}
			This endpoint allows the admin to display all the available restaurants as well as individual restaurant details .

/merchant/merchant-reviews	POST	{
		"op": "0",
		"merchant_id": "M-20230419101828",
		"review_id": "20230419104030",
		"adminO": "1"
		}
			This endpoint allows admin to view all store reviews and reviews of particular store and also allow admin to take actions on reviews.

/merchant/admins-list	POST	This request does not have body	This endpoint display list of admins

/merchant/register-merchant	POST	{
		"op": "0",
		"m_name": "Paradise Biryani",
		"m_latitude": "17.442946974044382",
		"m_longitude": "78.35687726975912",
		"m_city": "hyderabad",
		"m_street": "Gachibowli",
		"m_area": "anjai nagar",
		"m_pincode": "500032",
		"m_open_time": "11:00",
		"m_close_time": "23:00",
		"m_min_price": 300,
		"m_max_price": 1500,
		"m_specials": "Biryani",
		"m_categories": "veg&Non veg",
		"m_cusines": "north&south&chinese&biryani",
		"m_experience": "1958",
		"m_services": "takeaway&dine-in",
		"m_address": "Plot no 22 & 23, Gachibowli Rd, Vinayak Nagar, Indira Nagar, Gachibowli, Hyderabad, Telangana 500032"
		}
			This endpoint allow admin to add the restaurant to the DynamoDB database by verifying each and every individual fields.

4. RESULTS AND DISCUSSIONS

4.1 User Module

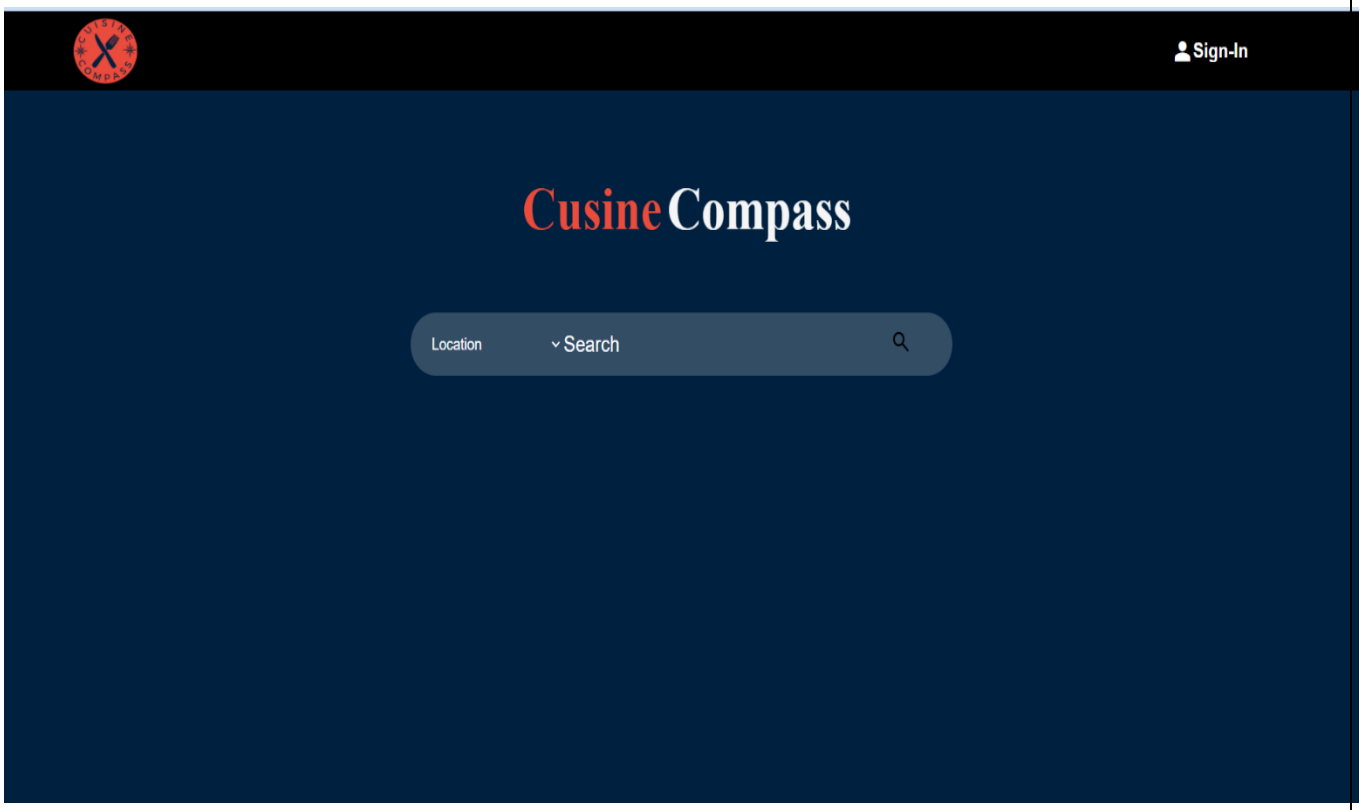


Fig. 4.1.1 Default Page

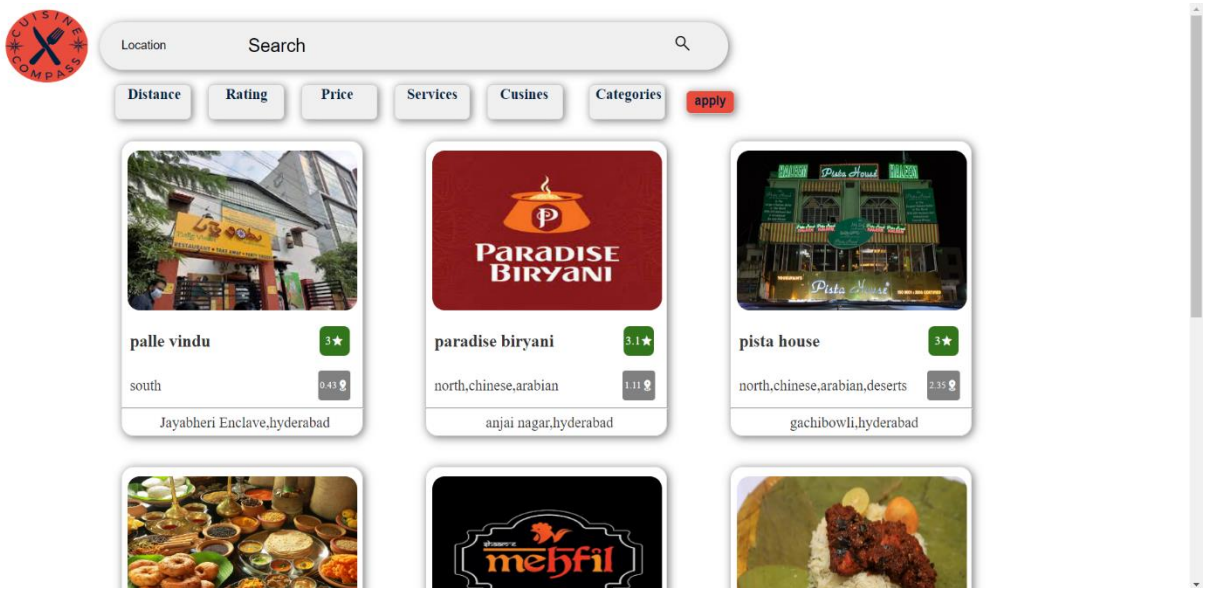


Fig. 4.1.2 Search Result Page when user type hotels near me

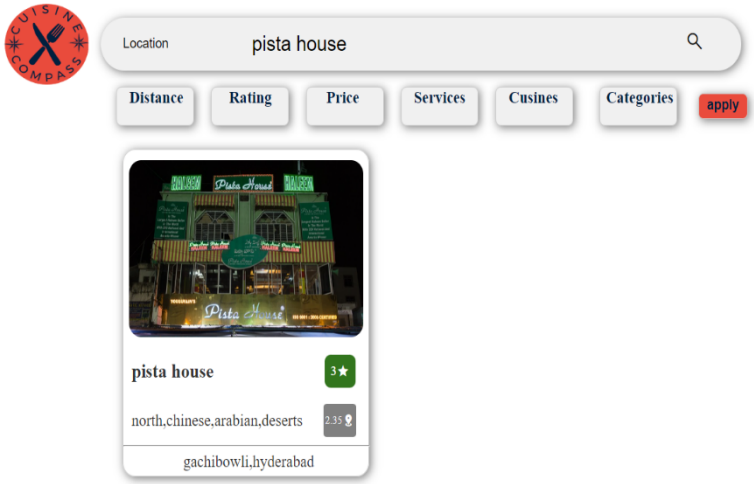


Fig. 4.1.3 Search Result when user input is restaurant name

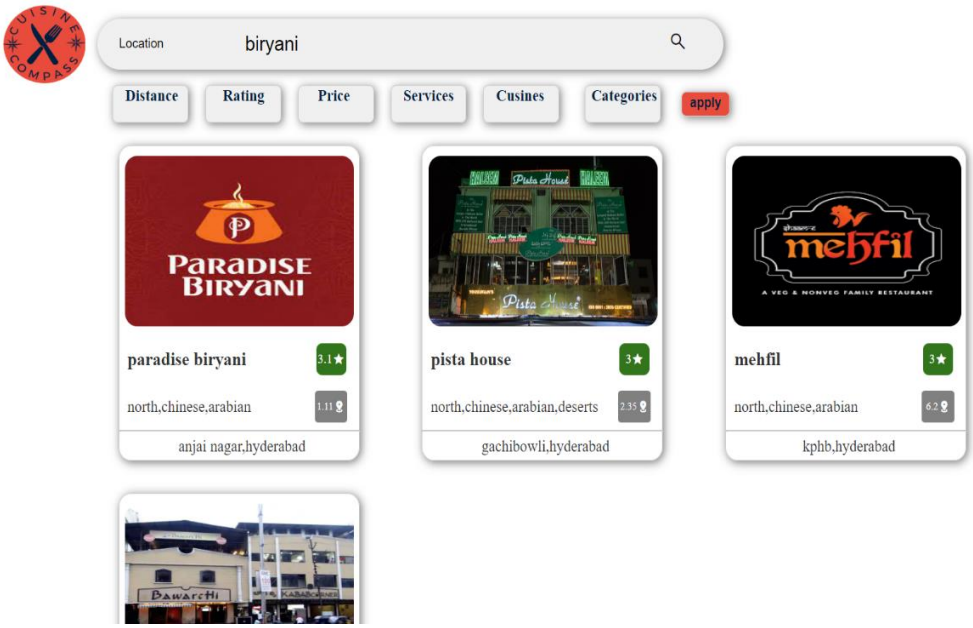


Fig. 4.1.4 Search Result when user input is dish name

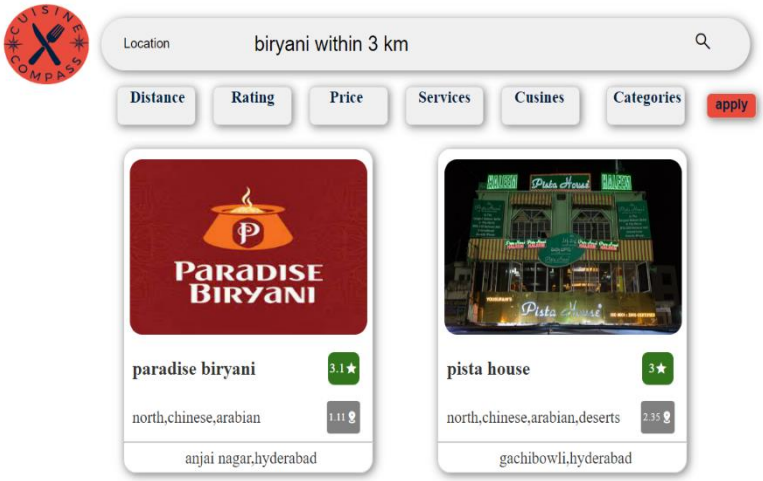


Fig. 4.1.5 Search Result when user input is dish name and distance

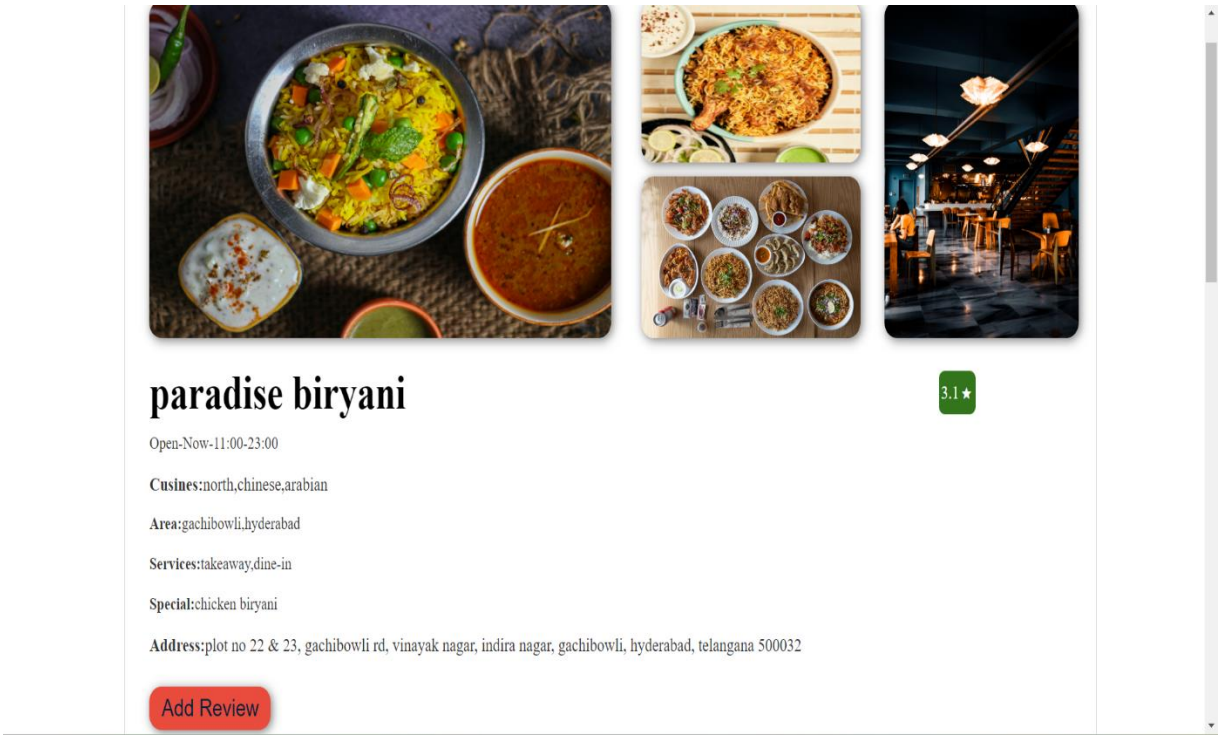


Fig. 4.1.6 Details of particular restaurant

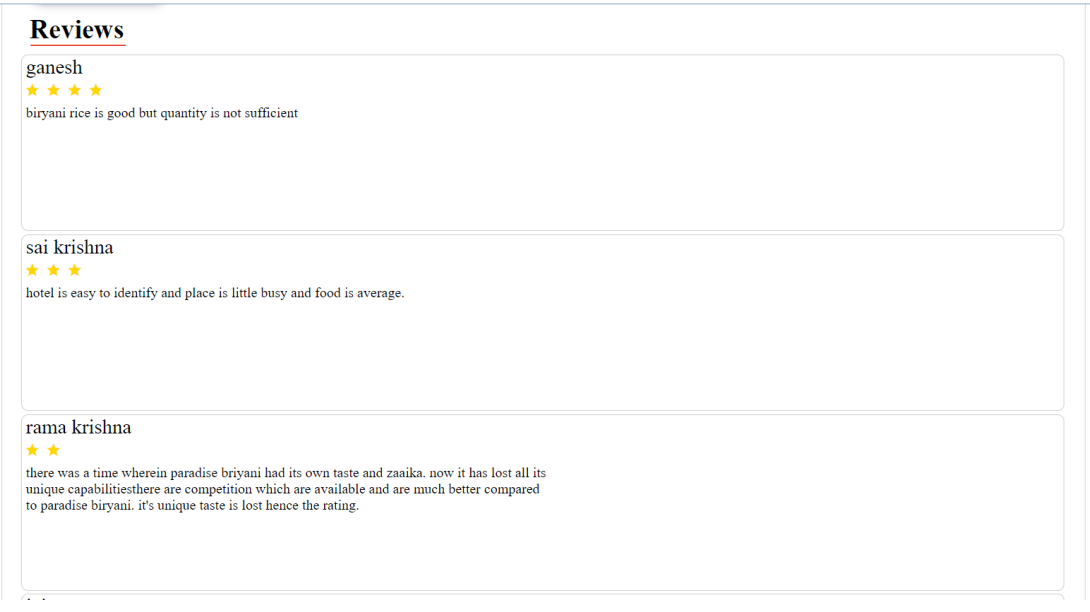


Fig. 4.1.7 Reviews of particular restaurant

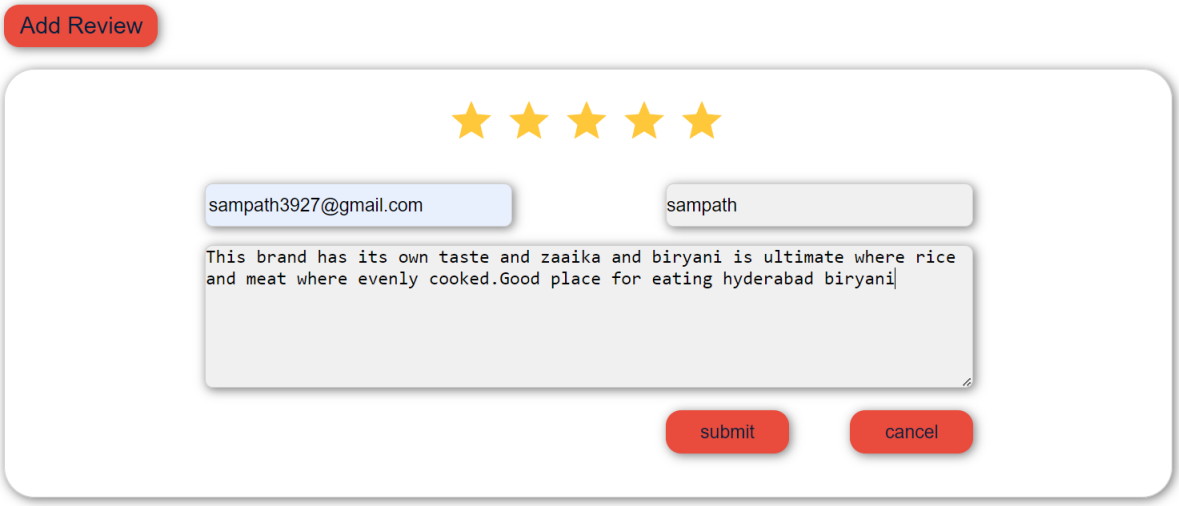


Fig. 4.1.8 Sample Review Form

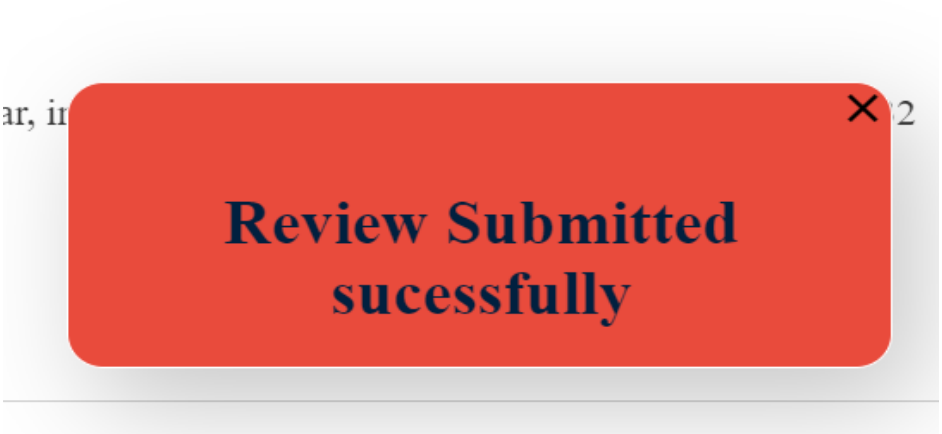


Fig. 4.1.9 Message after submitting the review

4.2 Admin Module

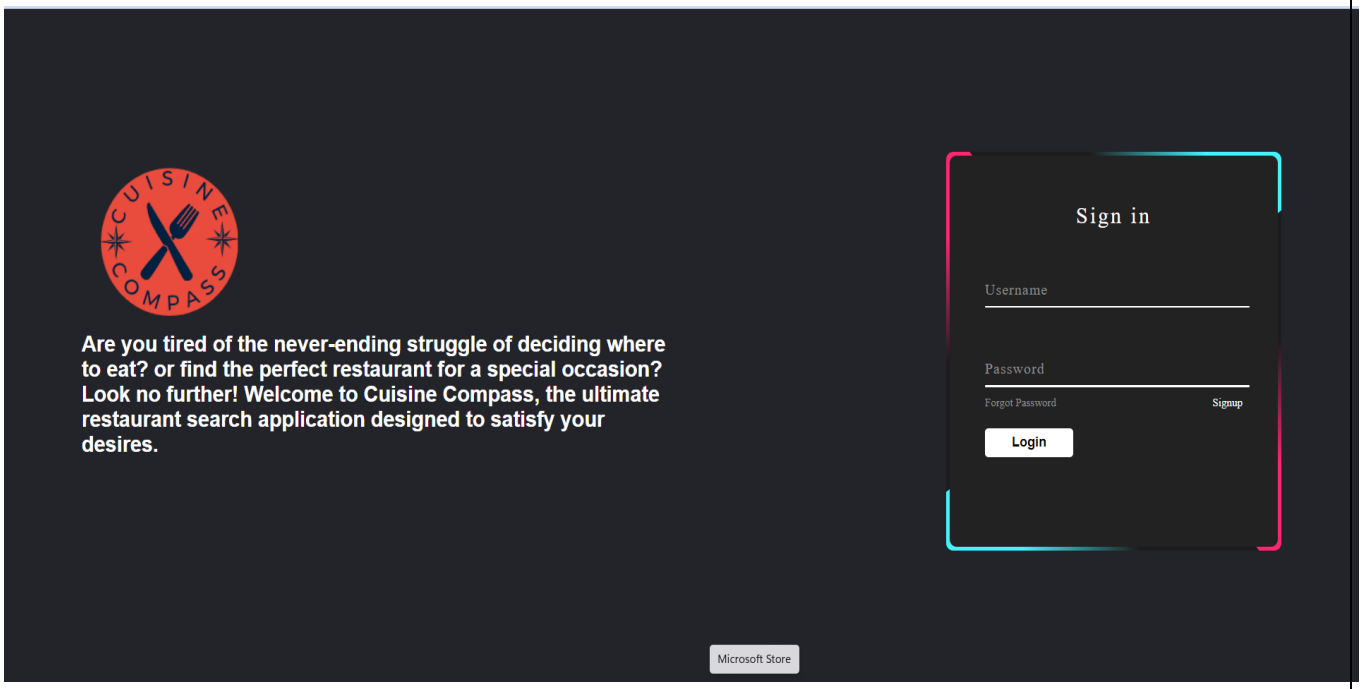


Fig. 4.2.1 Admin Login Page

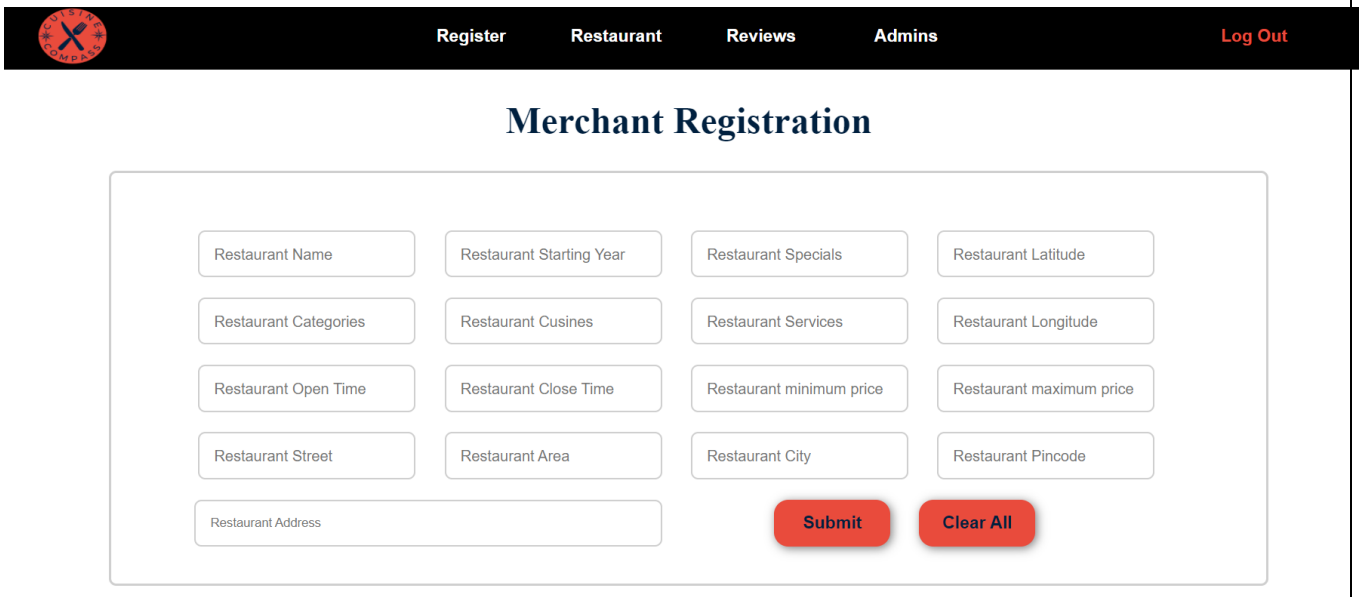
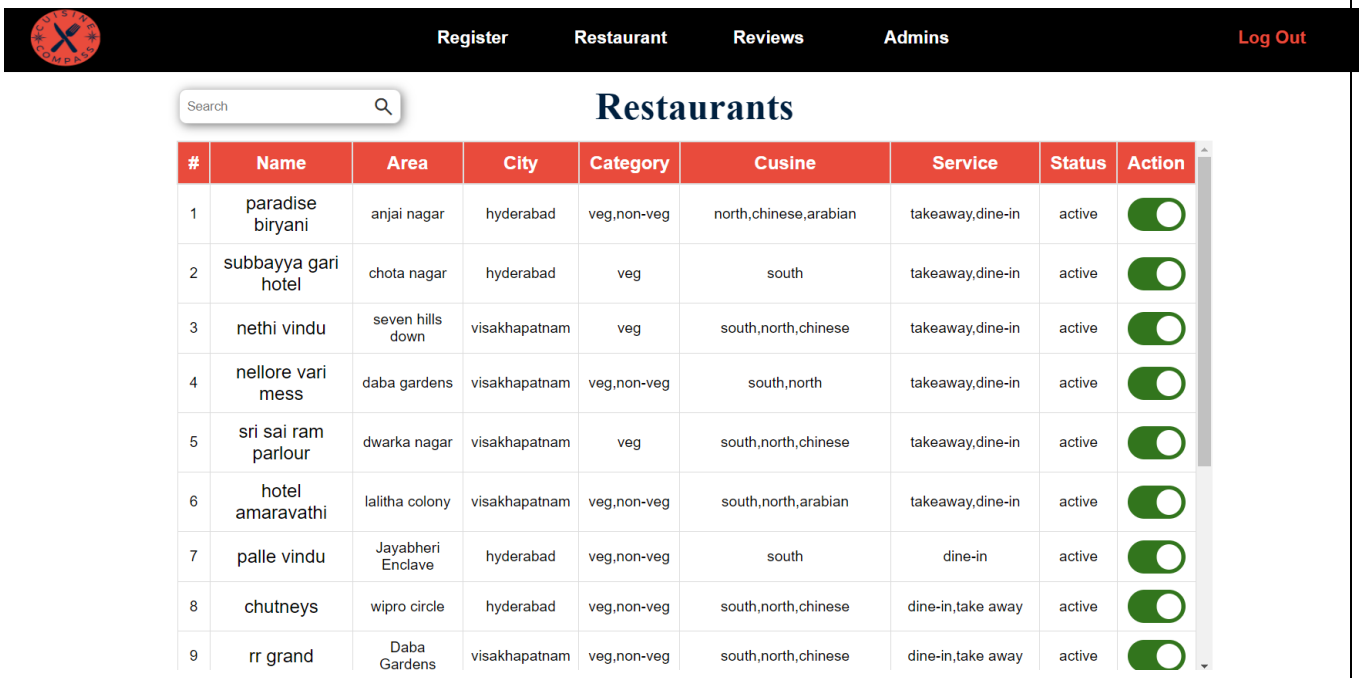
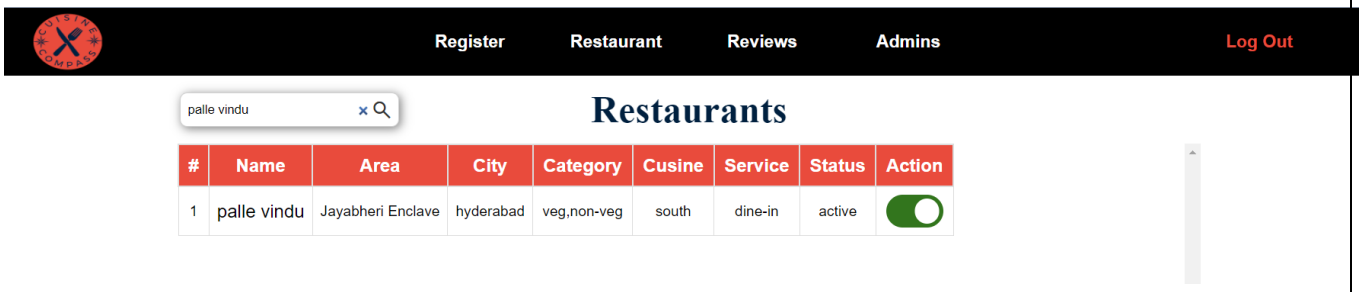


Fig. 4.2.2 Restaurant Registration



#	Name	Area	City	Category	Cusine	Service	Status	Action
1	paradise biryani	anjai nagar	hyderabad	veg,non-veg	north,chinese,arabian	takeaway,dine-in	active	
2	subbayya gari hotel	chota nagar	hyderabad	veg	south	takeaway,dine-in	active	
3	nethi vindu	seven hills down	visakhapatnam	veg	south,north,chinese	takeaway,dine-in	active	
4	nellore vari mess	daba gardens	visakhapatnam	veg,non-veg	south,north	takeaway,dine-in	active	
5	sri sai ram parlour	dwarka nagar	visakhapatnam	veg	south,north,chinese	takeaway,dine-in	active	
6	hotel amaravathi	lalitha colony	visakhapatnam	veg,non-veg	south,north,arabian	takeaway,dine-in	active	
7	palle vindu	Jayabheri Enclave	hyderabad	veg,non-veg	south	dine-in	active	
8	chutneys	wipro circle	hyderabad	veg,non-veg	south,north,chinese	dine-in,take away	active	
9	rr grand	Daba Gardens	visakhapatnam	veg,non-veg	south,north,chinese	dine-in,take away	active	

Fig. 4.2.3 Admin Restaurant Dashboard



#	Name	Area	City	Category	Cusine	Service	Status	Action
1	palle vindu	Jayabheri Enclave	hyderabad	veg,non-veg	south	dine-in	active	

Fig. 4.2.4 Searching Particular Restaurant in admin side

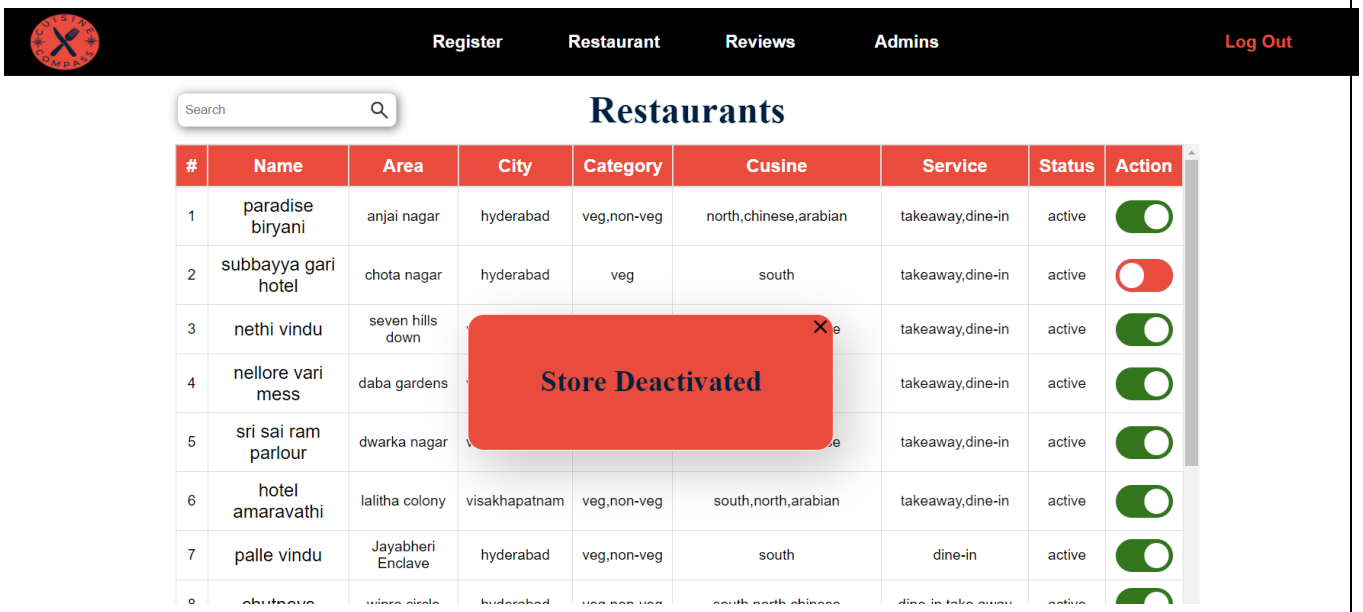


Fig. 4.2.5 Message Displayed when admin sets store to offline

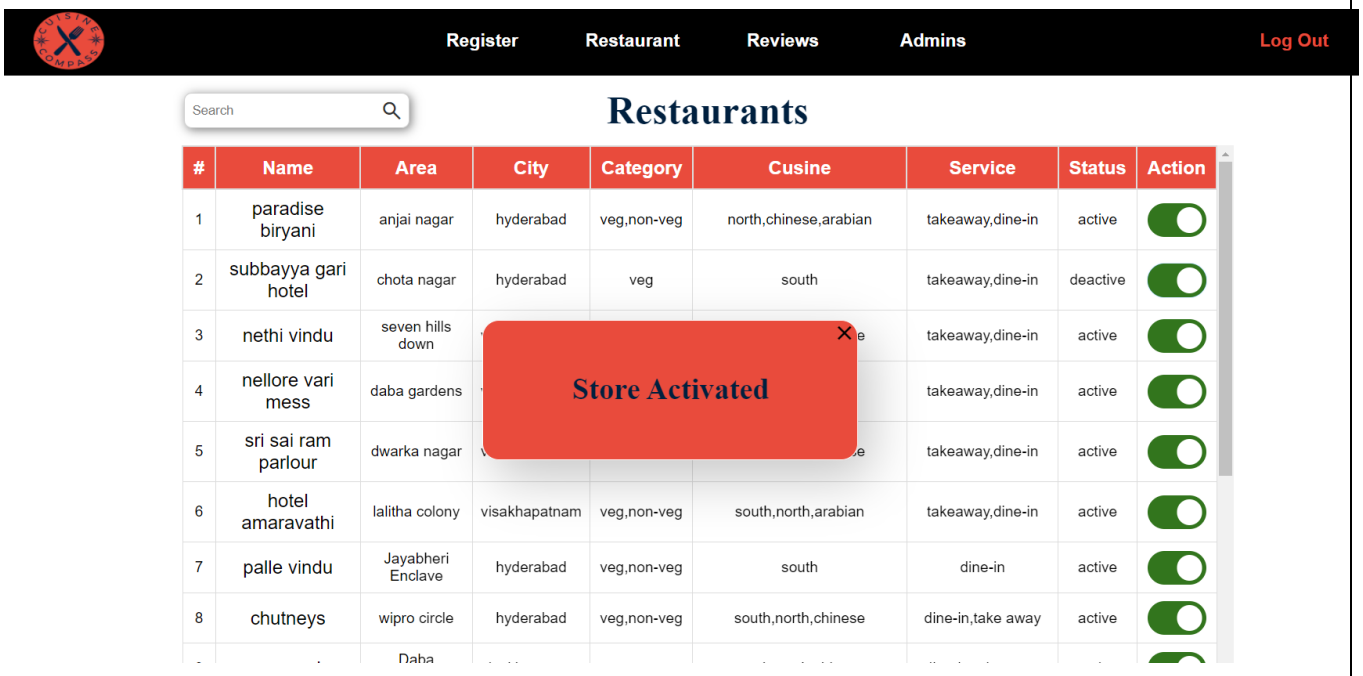


Fig. 4.2.6 Message Displayed when admin sets store to o

Fig. 4.2.7 Admin Side Restaurant Details

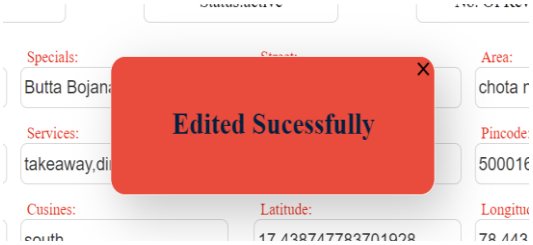


Fig. 4.2.8 Message displayed when admin edited the fields

Fig. 4.2.9 Admin Side Review Section

Fig. 4.2.10 Message displayed when admin approved the user reviews

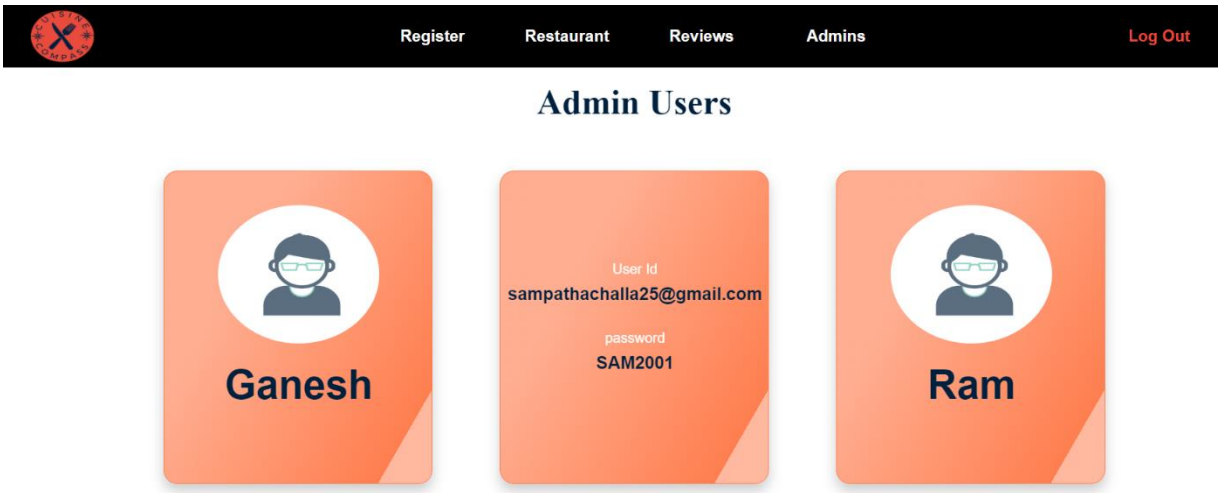


Fig. 4.2.11 List of Admin

5. TESTING AND VALIDATION

5.1 Frontend validations

Test Scenario: Search functionality

Test Case Id	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
TC 1.1	Check response when restaurant name is provided and it is available in the database	Search: Paradise Biryani	Able to display the paradise biryani in nearby location and user able to view details by single click.	Paradise biryani card appeared and by clicking on that it display details of the restaurant	Pass
TC 1.2	Check response when invalid restaurant name is given	Search: Train65	No restaurant results found prompt should be appeared on the screen	No restaurant results found prompt should be appeared on the screen	Pass
TC 1.3	Check response when dish name is supplied	Search: Idly	Able to display the restaurants which serve idly and south Indian dishes in nearby areas.	Displayed restaurants which serve idly in nearby locations.	Pass

TC 1.4	Check response when cuisine name is provided	Search: Italian	Able to display the restaurants which serve Italian dishes in nearby areas.	Displayed restaurants which serve Italian dishes in nearby locations.	Pass

TC 1.5	Check when Distance and rating is given	Search: Hotels with 3star rating in 5Kms distance	Able to display the restaurants which has 3 star ratings in 5 km radius	Displayed restaurants which has 3 star rating in 5 km distance	Pass
TC 1.6	Check response when filters are applied along with search input	Search: Hotels near me Filter: 4 star and veg	Able to display the restaurants which is under 50 kilometers distance and has 4 star and above rating and serve vegetarian dishes.	Displayed restaurants which serve vegetarian dishes and has 4 star and above rating under 50 kilometers distance.	Pass
TC 1.7	Check response when price of the dishes are mentioned	Search: Restaurants between 400 to 800 range	Able to display the restaurants which serve dishes between 400 to 800 or below.	Displayed restaurants which serve dishes in the mentioned range by user	Pass

Table 5.1.1 Test cases of search functionality

Test Scenario: Login functionality

Test Case Id	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
TC 2.1	Try to login without account	anonymous username , anonymous password	Error message should get prompt. Shouldn't get redirected to the dashboard.	Error message showing got prompt. Dashboard is not rendered.	Pass
TC 2.2	Enter verified login credentials	registered username , registered password	Should get redirected to admin dashboard	Got redirected to the admin dashboard.	Pass
TC 2.3	Enter invalid credentials.	Registered username, wrong password. Wrong username, registered password. Wrong username, wrong password	Error message should get prompt. Shouldn't get redirected to the dashboard.	Error message showing "Invalid username or password" got prompt.	Pass

Table 5.1.2 Test cases of Login functionality

Test Scenario: Register Restaurant

Test Case Id	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
TC 3.1	Check response when complete restaurant details is entered	Paradise biryani,11:00-23:00,300-1500,biryani,dine-in,veg&non-veg,Gachibowli, 17.442946974044382, 78.35687726975912	Form will be submitted. Will display message: Restaurant Registered Successfully.	Form will is submitted. And displayed message: Restaurant Registered Successfully.	Pass
TC 3.2	Check response when complete restaurant details is entered and it is already there	Paradise biryani,11:00-23:00,300-1500,biryani,dine-in,veg&non-veg,Gachibowli, 17.442946974044382, 78.35687726975912	Form won't get submitted. Will display message: Restaurant already existed	Form didn't get submitted and prompt the error message: Restaurant already existed	Pass
TC 3.3	Check response when required fields are not mentioned	Paradise biryani,11:00-23:00,300-1500,biryani,dine-in,veg&non-veg,Gachibowli,	Message should display: please check all the required fields	Expected Message gets displayed.	Pass

TC 3.4	Check the response when alphabets are mentioned in latitude and longitude	Latitude: 17.44294697e10 Longitude: 78.356877269759 @#	Message should display: please check all the required fields	Message get display:please check all the required fields	Pass
TC 3.5	Check the response when any special characters are mentioned other then coma(,) and dot(.)	Cuisines: north&Chinese& arabian	Message should display: please check all the required fields	Message get display:please check all the required fields	Pass
TC 3.6	Check response when no fields are filled and submitted the form	No Input Data	Message should be prompted: Incomplete Payload	Message displayed: Incomplete Payload .	Pass

Table 5.1.3 Test cases of Register Restaurant functionality

Test Scenario: Update Restaurant

Test Case Id	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
TC 4.1	Check when changing specials field	Specials: Shawarma	Message should be prompted: Edited Successfully	Expected Message gets displayed.	Pass
TC 4.2	Check when admin send store to offline	Click on deactivate button	Message should be prompted: Store Deactivated	Message prompted: Store Deactivated	Pass
TC 4.3	Check when admin accept the users reviews	Click on approved button	Message should be prompted: Review Accepted	Message prompted: Review Accepted	Pass
TC 4.4	Check when changing wrong fields with incorrect data	Pincode: 530	Message should be prompted: Error!	Expected Message gets displayed.	Pass

Table 5.1.4 Test cases of Update book functionality

Test Scenario: Add Review Functionality

Test Case Id	Test Case Description	Test Data	Expected Result	Actual Result	Pass /Fail
TC 5.1	Check response when proper review Is submitted	Name: Ganesh Rating:3 star Email:gani235 @gmail.com Review: Best	Message should be prompted: Review Submitted	Message prompted: Review Submitted	Pass
TC 5.2	Check response when valid name is not provided	Name: 123Krishna	Message should be prompted: Wrong Input check all the fields	Message prompted: Wrong Input check all the input fields	Pass
TC 5.3	Check response when Incomplete Review is submitted	Name: Sampath Rating: 4 star	Message should be prompted: Please fill all the required fields.	Expected Message gets displayed.	Pass
TC 5.4	Check response when star rating is not mentioned	Name: Ram Rating: Email:ram555 @gmail.com Review: budget friendly	Message should be prompted: Please fill all the required fields.	Message prompted: Please fill all the required fields	

Table 5.1.5 Test cases of Update Profile functionality

Backend Testing:

6. SUMMARY AND CONCLUSION

6.1 Summary of achievements

The achievements of a Restaurant search application(Cuisine Compass) project include:

The restaurant search application is a valuable tool that enhances the dining experience for users. It provides a convenient and efficient way to discover, explore, and select restaurants based on various preferences and criteria. It simplifies the process of finding suitable dining options by providing a centralized platform where users can access a vast database of restaurants. It eliminates the need for manual research and saves time and effort.

The application allows users to tailor their search based on preferences such as cuisine type, price range, location, and user ratings. This personalized approach ensures that users find restaurants that align with their individual tastes and requirements. This application provides detailed restaurant information, including specials, operating hours, services offered, and user reviews. This comprehensive information empowers users to make well-informed decisions and choose restaurants that meet their expectations.

Overall, this restaurant search application simplifies the process of discovering and selecting restaurants. By leveraging technology and user-generated content, this application enhances the overall dining experience and serves as a valuable tool for casual diners and food enthusiasts.

6.2 Main difficulties encountered

I faced issues while dealing with the review section. The review system was designed for admins in such a way that it can show all the user's reviews received for all the restaurants in the database as well as reviews received for the particular restaurant. As a traditional design, I kept merchant as the primary key and customer ID as the secondary key. With this design, we can retrieve the user reviews of particular restaurants for less cost. But getting the total number of user reviews of all the restaurants becomes costly because it has to loop the database twice, and it will stall the servers as the restaurant count in the database increases. To solve this issue, we use the indexing property of the dynamoDb table. By using this property, I can retrieve all the user reviews at less cost.

I faced issues while dealing with the search integration because I need this system in such a way that users can get accurate results in every scenario they use, which means providing the restaurant name (or) location (or), distance (or) pricing (or) multiple fields at once. Searching with the restaurant name is simple, but it will be challenging if they use other scenarios. For that, I created different flags. Whenever the flags trigger, the algorithm will be able to notify the user input, which triggers the flag and store the information in the filter list. The restaurant list will be generated accordingly.

6.3 Limitations of the project

This application will slow down if the number of restaurants in particular areas increases because this application will take distance as the primary attribute of selection. Suppose the number of restaurants increases for a specific location. In that case, the number of restaurant lists will increase dynamically, and the system will take lots of time to process the list and generate the filtered list for the user. To solve this type of problem, we need more accurate location tracking and unique address mapping. However, such a system would require additional resources and infrastructure, which may only be practical for some instances.

This application needs to be fully automated in verifying user reviews, and it faces difficulties if there are more and more incoming reviews and the admin or responsible person needs more time to verify all the user reviews. It will be stalled entirely if the responsible person is out of reach. Since the human intervention is present for verifying the reviews, it is also prone to errors because some times admin will allow fake reviews, which causes the system to lose its reliability.

6.4 Future Scope of work

In the future, Adding more filter fields to the system is better, which will increase the system's efficiency. It is better to fully automate the user review system by using Artificial intelligence to replace the human work of reviewing the user reviews. Using AI will make the system less prone to errors and can be used for further studies. Machine learning algorithms can help the application understand the users' preferences over time and provide tailored restaurant recommendations to the users based on their previous search history in the application.

Integrate social media features into the application, allowing users to share their restaurant experiences, photos, and reviews with friends and followers. Collaborate with food delivery platforms to offer seamless integration. Users can not only search for restaurants but also place orders for delivery or takeout without having to switch between multiple apps.

Multilingual support may be a significant development for this type of application. A more comprehensive range of users can access the system by adding support for several languages. This feature might boost utilization and user satisfaction by enabling users from various geographic and linguistic backgrounds to utilize the system in the most comfortable way. The application may become more inclusive and varied by attracting users from non-native English-speaking nations with the aid of multilingual assistance.

6.5 Conclusion

The restaurant search application allows users to quickly find and explore various dining options in their specific location. Users can quickly search for restaurants based on their preferences by applying the filters provided. They save time and effort compared to traditional methods like asking for recommendations or physically visiting multiple establishments.

The application database has a wide range of budget-friendly restaurants and hidden gem stores, which may be excluded during the traditional search. Users can access various dining options catering to different tastes, budgets, and preferences. This variety allows users to discover new and diverse culinary experiences.

This application often features user-generated reviews and ratings. Users can read feedback from other diners to make informed decisions about which restaurants to visit. This will help users gauge the quality of the dining experience and make choices based on the opinions of others.

By using this application, users can support local and independent restaurants. This application often features a mix of well-known chains and local establishments, helping users discover and contribute to their local dining scene.

In summary, restaurant search applications provide convenience, a wide selection of choices, user reviews and ratings, real-time information, location-based recommendations, special offers, seamless integration, personalization, opportunities for discovering new restaurants, and support for local businesses. These advantages enhance the overall dining experience and make it easier for users to find and enjoy great meals.

7. REFERENCES

[1] Aureliano-Silva, Leonardo & Leung, Xi & Spers, Eduardo Eugênio. (2021). The effect of online reviews on restaurant visit intentions: applying signaling and involvement theories. Journal of Hospitality and Tourism Technology. ahead-of-print. 10.1108/JHTT-06-2020-0143.

[2] Kim, Jaewook & Lee, Minwoo & Kwon, Wooseok & Park, Hyekyung & Back, Ki-Joon. (2022). Why am I satisfied? See my reviews – Price and location matter in the restaurant industry. International Journal of Hospitality Management. 101. 103111. 10.1016/j.ijhm.2021.103111.

[3] Isabela, Erika & Drona, Jennifer & Fadhilah, Nailatul & Tanoto, Dian & Harefa, Jeklin & Prajena, Gredion & Chowanda, Andry & Alexander,. (2018). NYAM: An Android Based Application for Food Finding Using GPS. Procedia Computer Science. 135. 393-399. 10.1016/j.procs.2018.08.189.