

## **TABLE OF CONTENTS**

TABLE OF CONTENTS.....	1
1. SYNOPSIS .....	1
1.1.Title of the Project .....	2
1.2.Abstract.....	2
1.3.Objective of the Project .....	2
1.4.Project Category.....	3
1.5.Language(s) to be used.....	3
1.5.1. Front-end .....	3
1.5.2. Back-end.....	3
1.5.3. Database :.....	3
1.6.Structure of the proposed project.....	3
1.7.Module Description .....	4
1.7.1. User Management Module:.....	4
1.7.2. Restaurant Discovery Module:.....	4
1.7.3. User Review Module:.....	4
1.7.4. Favourites Management Module: .....	4
1.7.5. Drops Module: .....	4
1.7.6. Price Filtering Module:.....	5
1.7.7. Admin Panel Module:.....	5
1.8.Future scope of the Project .....	5
1.9.Bibliography.....	6
2. SOFTWARE REQUIREMENTS AND SPECIFICATION .....	7

2.1. Introduction .....	8
2.1.1. Purpose .....	8
2.1.2. Intended Audience and Reading Suggestions .....	8
2.1.3. Product Scope .....	9
2.1.4. References .....	9
2.2. Overall Description.....	10
2.2.1. Product Perspective .....	10
2.2.2. Product Functions .....	10
2.2.3. User Classes and Characteristics .....	11
2.2.4. Operating Environment .....	11
2.2.5. Assumptions and Dependencies .....	12
2.3. External Interface Requirements.....	12
2.3.1. User Interfaces .....	12
2.3.2. Hardware Interfaces.....	12
2.3.3. Software Interfaces .....	13
2.3.4. Communications Interfaces.....	13
2.4. System Features .....	14
2.4.1. User Registration and Authentication.....	14
2.4.2. Geo-location:.....	14
2.4.3. Restaurant information .....	15
2.4.4. Reviews: .....	15
2.4.5. Favorites Management: .....	15
2.4.6. Drops (Food Vlog Upload): .....	16
2.4.7. Price Filtering:.....	16

2.4.8. Restaurant Management (Admin Panel): .....	16
2.5. Other Nonfunctional Requirements .....	17
2.5.1. Performance Requirements .....	17
2.5.2. Security Requirements.....	17
2.5.3. Software Quality Attributes .....	17
2.6.1. LEVEL 0 DFD OR (CONTEXT FLOW DIAGRAM) .....	19
2.6.2. LEVEL 1 DFD (ADMIN) .....	19
2.6.3. LEVEL 1 DFD (USER) .....	20
3 System Modeling .....	<b>Error! Bookmark not defined.</b>
3.1 Class Diagram .....	<b>Error! Bookmark not defined.</b>
3.2 USECASE .....	25
3.3 SEQUENCE DIAGRAM.....	28
4 DATABASE MODELING .....	32
4.1 ER DIAGRAM.....	33
4.2 TABLE DESCRIPTION .....	36
5. TESTING AND VALIDATION .....	41
5.1. INTRODUCTION .....	42
5.2. TESTING STEPS .....	42
5.2.1. Unit Testing.....	42
5.2.2. Integration Testing .....	42
5.2.3. Validation .....	43
5.2.4. Output testing .....	43
5.2.5. User acceptance testing .....	43
5.3. Test Cases and Validation .....	<b>Error! Bookmark not defined.</b>

5.3.1. Sign up form.....	44
5.3.2. Login form.....	48
5.3.3. Home page.....	52
5.3.4. Submit restaurant .....	56
5.3.5. Hotel page .....	59
5.3.6. Item page .....	62
5.3.7. Drops page.....	65
5.3.8. Upload drop window .....	67
5.3.9. User profile.....	69
5.3.10. Admin profile.....	73
5.3.11. Admin panel.....	74
5.3.12. Add new restaurant form.....	76
5.3.13. Manage restaurant form.....	78
5.3.14. Add item form.....	80
5.3.15. User suggestion review .....	82
CONCLUSION .....	84

## **1. SYNOPSIS**

## **1.1.Title of the Project**

Eatables

## **1.2.Abstract**

EATABLES is a location-based food discovery platform that allows users to explore and find new dining options. Users can access information about nearby restaurants, including contact details, location, and reviews from other users. They can leave their own reviews, add food items to favorites, and upload food vlog videos through the "drops" feature. The platform offers filtered search results based on price and allows users to submit new restaurant additions for review by the admin. EATABLES aims to provide a comprehensive and engaging experience for users to discover, share, and review food options in their area.

## **1.3.Objective of the Project**

The objective of the EATABLES project is to provide users with a convenient and comprehensive platform to discover new food options based on their location. The project aims to fulfill the following objectives:

- Location-based Food Discovery: Enable users to find nearby restaurants, view their information, and read reviews from other users, helping them make informed dining choices.
- User Reviews and Engagement: Allow users to leave their own reviews, fostering a community-driven approach and facilitating knowledge sharing about dining experiences.
- Favorites List: Provide users with the ability to create a personalized list of favorite food items or restaurants for easy reference and future visits.
- Drops Feature: Allow users to upload and share food vlog videos, enhancing engagement and providing a platform for users to discover and appreciate food-related content
- Price Filtering: Offer users the option to filter search results based on price, ensuring that they can find dining options that align with their budget or desired price range.

- Restaurant Submission: Enable users to suggest new restaurants to be added to the platform, ensuring the continuous growth and relevance of the database.

Overall, the objective of the EATABLES project is to enhance the dining experience for users by providing them with a user-friendly platform that leverages location data, user reviews, engagement features, and personalized preferences to facilitate food discovery and decision-making.

## **1.4. Project Category**

Web-based Application

## **1.5. Language(s) to be used**

**1.5.1.Front-end :** HTML, CSS & JavaScript

**1.5.2.Back-end :** PHP

**1.5.3.Database :** MySQL

## **1.6. Structure of the proposed project**

- User Management: This component enables user authentication and registration, allowing users to create accounts and log in to access the system.
- Location-Based Restaurant Discovery: Using the device's location services, this component identifies nearby restaurants and provides users with relevant information, such as contact details and location.
- User Reviews: Users can leave reviews for restaurants they have visited, sharing their experiences and opinions to help other users make informed dining decisions.
- Favorites Management: Users can create a personalized list of favorite food items or restaurants for easy access and future reference.
- Drops: The Drops feature allows users to upload and share food vlog videos, fostering engagement and enabling users to discover and appreciate food-related content.
- Price Filtering: Users can filter search results based on their desired price range, aiding in finding dining options that match their budget.
- Restaurant Management (Admin Panel): This component grants administrative

privileges and allows the admin to manage restaurants. It includes functionalities such as adding new restaurants, editing existing ones, and reviewing user requests for adding new restaurants.

## **1.7.Module Description**

### **1.7.1.User Management Module:**

This module handles user authentication and registration. It allows users to create accounts, log in securely, and manage their profiles.

### **1.7.2.Restaurant Discovery Module:**

The Restaurant Discovery module utilizes the device's location services to identify nearby restaurants. It provides users with essential information such as contact details, location, and user reviews.

### **1.7.3.User Review Module:**

The User Review module enables users to leave reviews for restaurants they have visited. Users can share their dining experiences, providing valuable insights for other users.

### **1.7.4.Favourites Management Module:**

With the Favourites Management module, users can create a personalized list of their favourite food items or restaurants. This module allows users to easily access and organize their preferred options.

### **1.7.5.Drops Module:**

The Drops module is designed for users to upload and share food vlog videos. Users can showcase their culinary experiences and engage with other users' content.

#### **1.7.6.Price Filtering Module:**

The Price Filtering module enables users to filter search results based on their desired price range. This module helps users find dining options that fit their budget.

#### **1.7.7.Admin Panel Module:**

The Admin Panel module is accessible only to the admin. It allows the admin to manage restaurants, including adding new restaurants, editing existing ones, and reviewing user requests for adding new restaurants.

### **1.8.Future scope of the Project**

- **Integration of Advanced Recommendation Systems:** The platform can incorporate advanced recommendation algorithms to provide personalized food recommendations to users based on their preferences, previous reviews, and browsing history. This can greatly enhance the user experience and help users discover new food options tailored to their individual tastes.
- **Social Media Integration and Influencer Collaborations:** The project can explore integration with popular social media platforms, enabling users to share their dining experiences and reviews on their social media accounts directly from the EATABLES platform. Collaborations with food influencers or bloggers can also be considered to enhance the visibility and engagement of the platform.
- **Enhanced User Engagement and Gamification:** Implementing gamification elements such as badges, rewards, and challenges can boost user engagement and create a more interactive experience. Users can earn rewards for leaving reviews, participating in community activities, or achieving specific milestones within the platform.
- **Expansion to Additional Geographical Locations:** While initially focusing on a specific location, the project can expand its reach to cover more cities or even different countries, allowing users to discover food options in various locations.
- **Integration with Third-Party Services:** The platform can explore partnerships and integrations with third-party services such as reservation systems, food delivery aggregators, or restaurant booking platforms. This would offer users a comprehensive solution for discovering, reviewing,

## **1.9.Bibliography**

- LECTURE NOTES ON SOFTWARE ENGINEERING

By Dr. H.S.Behera Asst. Prof K.K.Sahu Asst. Prof Gargi Bhattacharjee

- Fundamentals of Software Engineering, PHI

By Mall Rajib

## **2. SOFTWARE REQUIREMENTS AND SPECIFICATION**

## **2.1.Introduction**

The Software Requirements Specification (SRS) document is a crucial artifact in the software development process. It serves as a comprehensive guide that captures the functional and non-functional requirements of a software system. The SRS document outlines the purpose, scope, and architecture of the software, providing a clear understanding of what needs to be developed and how it should behave. It serves as a reference for stakeholders, including developers, designers, testers, and clients, ensuring a common understanding of the project goals and facilitating effective communication and collaboration throughout the software development lifecycle. The SRS document acts as a contract between the development team and the stakeholders, guiding the development process and serving as a benchmark for validation and verification of the final product.

### **2.1.1.Purpose**

The purpose of the project is to develop a comprehensive and user-friendly food discovery platform that allows users to explore, review, and interact with restaurants in their vicinity. The project aims to simplify the process of finding new dining options, providing users with valuable information, user reviews, and the ability to save favorites. By creating a platform that enhances the dining experience and facilitates informed decision-making, the project seeks to improve user satisfaction and promote the culinary exploration of different food options.

### **2.1.2.Intended Audience and Reading Suggestions**

- Project team members: Students who are working on the project, including developers, designers, and testers.
- Internal guide: The SRS will be reviewed and evaluated by the internal guide who is overseeing the project.
- Peers: Other students in the class may also review the SRS to provide feedback and suggestions.
- External guide: They will want to review the document to ensure that the product meets their needs and expectations.

Reading suggestions for the SRS document could include the following:

- Review the document in its entirety to get a general understanding of the project and its requirements.
- Focus on specific sections that are relevant to readers role on the project, such as the functional requirements or non-functional requirements.
- Review the document periodically throughout the project to ensure that the requirements are being met and to make any necessary updates.

### **2.1.3. Product Scope**

The platform aims to provide users with a convenient and user-friendly interface to discover nearby restaurants, view essential information such as contact details and location, and read reviews from other users. Users can also leave their own reviews, add food items or restaurants to their favorites list, and upload food vlog videos through the Drops feature. Additionally, the project includes an admin panel for restaurant management, allowing the admin to add, edit, or delete restaurants, manage menu items, and review user requests for adding new restaurants.

The project focuses on delivering the core functionalities mentioned while ensuring a seamless user experience and providing a comprehensive solution for food discovery and engagement.

### **2.1.4. References**

- IEEE Software Requirements Specification Template:  
[https://web.cs.dal.ca/~hawkey/3130/srs\\_template-ieee.doc](https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc)
- How to Write a Software Requirements Specification (SRS Document):  
<https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document>
- 6 Steps for Writing an SRS That Works:  
<https://www.uptech.team/blog/srs-document>

## **2.2.Overall Description**

### **2.2.1.Product Perspective**

The EATABLES food discovery platform is designed to operate as a standalone system, independent of any other software or external systems. It serves as a centralized platform for users to explore and engage with restaurants in their vicinity. While the platform interacts with external services such as location services for restaurant discovery, it does not rely on or integrate with any specific external systems for its core functionality.

The platform is developed as a web-based application accessible through web browsers on various devices, including desktops, laptops, tablets, and smartphones. It does not require any specific hardware components or dependencies beyond standard web-browsing capabilities.

As an independent system, EATABLES aims to provide a comprehensive solution for users to discover, review, and interact with restaurants. It does not seek to replace or integrate with existing restaurant management systems or third-party services. However, future integrations with external systems such as online ordering and delivery services can be considered to enhance the user experience and expand the platform's capabilities

### **2.2.2.Product Functions**

- **User Registration and Authentication:** Users can create accounts and securely log in to access the platform's
- **Restaurant Discovery:** The platform utilizes location services to identify nearby restaurants and provides users with essential information, such as contact details, location, and reviews from other users.
- **User Reviews:** Users can leave detailed reviews for restaurants they have visited, sharing their experiences and opinions to assist other users in making informed dining decisions.
- **Favorites Management:** Users can create a personalized list of favorite food items or restaurants, allowing for quick and easy access to their preferred options.

- Drops (Food Vlog Upload): Users have the ability to upload and share food vlog videos, creating an engaging platform for users to showcase their culinary experiences and interact with others.
- Price Filtering: Users can filter search results based on their desired price range, helping them find dining options that align with their budget or affordability preferences.
- Admin Panel: The admin panel provides administrative privileges, allowing the admin to manage restaurants, add new restaurants, edit existing ones, and review user requests for adding new restaurants.

### **2.2.3.User Classes and Characteristics**

- General Users: These are individuals who use the platform to discover and explore restaurants in their vicinity. They may have varying levels of technological expertise and preferences when it comes to dining experiences. General users rely on the platform to provide accurate and relevant restaurant information, user reviews, and an intuitive interface for seamless navigation.
- Administrators: Administrators have elevated privileges and are responsible for overseeing the overall functioning of the platform. They have access to the admin panel to review user requests for adding new restaurants, moderate content, and maintain the integrity of the platform. Administrators possess advanced system management skills and a comprehensive understanding of the platform's functionalities.

### **2.2.4.Operating Environment**

- Hardware platform: the system should be accessible on both desktop and mobile devices, including laptops, smartphones, and tablets.
- Operating system: the system should be compatible with popular operating systems such as Windows, mac OS, iOS, and Android.
- Browser compatibility: the system should be accessible via modern web browsers, such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

## **2.2.5.Assumptions and Dependencies**

- Location data accuracy: The system relies on accurate location data to provide relevant results and recommendations. If the location data is not accurate, the system may provide incorrect results, affecting the user experience.
- User reviews: The system is based on user reviews, which may not always be accurate or trustworthy. There is a risk that fake reviews could be posted, affecting the accuracy of the information provided by the system.
- Dependency on network connectivity: The system may rely on network connectivity to retrieve data from the server and provide results to the user. If network connectivity is poor or unavailable, the system may not work properly.

## **2.3.External Interface Requirements**

### **2.3.1.User Interfaces**

The interface should be easy to navigate, with clear and consistent labeling and a logical hierarchy of information.

The system should display ratings and reviews for each restaurant, with the ability for users to leave their own reviews and ratings. The review system should be easy to use and understand, with clear guidelines for leaving reviews and feedback.

The system should include maps to help users find the restaurants they are interested in. Users should be able to view maps with clear markers for each restaurant. The system should be designed to be mobile responsive, with a layout that adapts to different screen sizes and resolutions. This will ensure that users can access your system from any device, whether they are using a desktop computer, tablet, or smartphone.

### **2.3.2.Hardware Interfaces**

Supported device types: The Eatables platform should support a range of device types, including desktop and mobile devices. This will involve optimizing the UI for different screen sizes and orientations, as well as ensuring compatibility with different operating systems (e.g., Windows, macOS, iOS, Android).

### **2.3.3. Software Interfaces**

- Databases: The Eatables platform will require a database to store and retrieve user data, such as reviews, ratings, and restaurant information. The database could be a SQL or NoSQL database, such as MySQL, MongoDB, or Cassandra, depending on the specific requirements of the platform.
- Operating systems: The Eatables platform should be compatible with a range of operating systems, including Windows, macOS, iOS, and Android.
- Data items and messages: The data items and messages coming into the system will primarily include user-generated data, such as reviews, ratings, as well as location data from the user's device. Data items and messages going out of the system will primarily include restaurant information, recommendations, and search results.
- Services needed: The Eatables platform will require a range of services to support its functionality, such as geolocation services, data storage and retrieval services, and user authentication services.
- Shared data: Data that will be shared across software components will primarily include user data, such as reviews, ratings, as well as restaurant information. The data sharing mechanism will likely involve database queries and API requests and responses.

### **2.3.4. Communications Interfaces**

- Web browser: The Eatables platform will primarily be accessed through a web browser on desktop and mobile devices. It should support modern browsers such as Google Chrome, Mozilla Firefox, and Safari.
- Electronic forms: The Eatables platform may use electronic forms for user authentication, such as sign up and login, and for collecting user-generated data, such as reviews and ratings.
- Communication standards: The Eatables platform may use HTTP or HTTPS for communication between the client and server. It may also use the Geolocation API for retrieving location data from the user's device.

## **2.4.System Features**

### **2.4.1.User Registration and Authentication**

Description:

This feature allows users to create an account and login securely to access the system.

Stimulus/Response Sequence:

User clicks on the "Sign Up" button on the login page

System prompts the user to enter their name, email address, and password

User fills in the details and clicks on the "Submit" button

System verifies the information and creates the user account

User enters their credentials and clicks on the "Login" button

System authenticates the user and grants access to the platform

Functional Requirements:

User account creation

Secure login and authentication process

### **2.4.2.Geo-location:**

Description:

This feature uses the device's location to provide users with a list of nearby restaurants. When a user opens the system, it should request permission to access the user's location. If the user grants permission, the system should display a list of restaurants near the user's current location.

Stimulus/Response Sequence:

User opens the app -> App requests permission to access the user's location -> User grants permission -> App displays a list of nearby restaurants.

Functional Requirements:

The system should use an accurate and reliable geolocation service to determine the user's location.

### **2.4.3.Restaurant information**

Description:

This feature displays detailed information about each restaurant, including its name, address, contact information, menu items, and reviews.

Stimulus/Response Sequence:

User selects a restaurant -> App displays detailed information about the restaurant.

Functional Requirements:

Location-based restaurant identification

Display of essential restaurant information

Filtering options based on price

### **2.4.4.Reviews:**

Description:

This feature allows users to leave reviews for each food items and read reviews posted by others.

Stimulus/Response Sequence:

User selects a restaurant -> User selects an item -->User leaves a review  
--> App displays the review.

Functional Requirements:

The system should have a clear and easy-to-use rating and review system.

### **2.4.5.Favorites Management:**

Description:

This feature allows users to create a personalized list of favorite food items or restaurants and quickly access them for future reference.

Stimulus/Response Sequence:

User selects a restaurant -> User selects an item --> user add item to their favorites list

User can access their list of favorites from their account page

Functional Requirements:

Option for users to create a personalized list of favorite food items

Quick access to saved favorites for future reference

#### **2.4.6.Drops (Food Vlog Upload):**

Description:

This feature allows users to upload and share their food vlog videos and interact with other users' videos.

Stimulus/Response Sequence:

User creates a food vlog video -->User uploads the video to the system --

>

System displays the video on the Drops page for other users to view and interact with.

Other users can like, comment, and share the video

Functional Requirements:

Users can upload and share their food vlog videos

Like and comment functionalities on uploaded videos

#### **2.4.7.Price Filtering:**

Description:

This feature allows users to filter search results based on their desired price range.

Stimulus/Response Sequence:

User selects a price range filter option -->System retrieves and displays the list of restaurants that fall within the selected price range

Functional Requirements:

Filtering search results based on desired price range

Helping users find dining options within their budget

#### **2.4.8.Restaurant Management (Admin Panel):**

Description:

This feature allows the admin to manage restaurants and their information, including adding new restaurants to the platform, editing or deleting existing restaurant details, and reviewing and approving user requests for adding new restaurants.

Stimulus/Response Sequence:

Admin logs into the admin panel using their credentials-->  
System authenticates the admin and grants access to the  
admin functionalities-->Admin manages the restaurant

Functional Requirements:

- Admin login and authentication
- Adding new restaurants to the platform
- Editing and updating existing restaurant details
- Deleting restaurants from the platform
- Reviewing and approving user requests for adding new restaurants

## **2.5.Other Nonfunctional Requirements**

### **2.5.1.Performance Requirements**

Speed: The app should be fast and responsive, with pages that load quickly and smoothly. Users expect web pages to load within a few seconds.

### **2.5.2.Security Requirements**

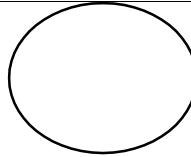
The web application should validate user input to prevent SQL injection attacks and cross-site scripting (XSS) attacks.

### **2.5.3.Software Quality Attributes**

Reliability: the app should be available and functional at all times, with minimal downtime or errors. It should also be able to recover quickly in the event of a failure or disruption.

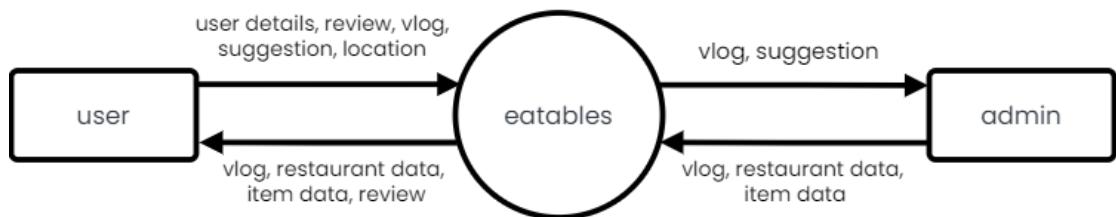
## 2.6.DATA FLOW DIAGRAM

A Data Flow Diagram is the graphical representation of the data through an information system, modelling its process aspects. A DFD also known as bubble chart or dataflow graph are commonly used during problem analysis. DFD'S are very useful in understanding a system and can be efficiently used during analysis. A DFD will show what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored it does not show information about process timing or whether process will operate in sequence or in parallel format, a DFD is often used as the preliminary step to create an overview of the system without going into great detail, which can later be elaborated, it can also be used for the visualization of data processing DFD shows the movement of data through different transformations or process in the system.

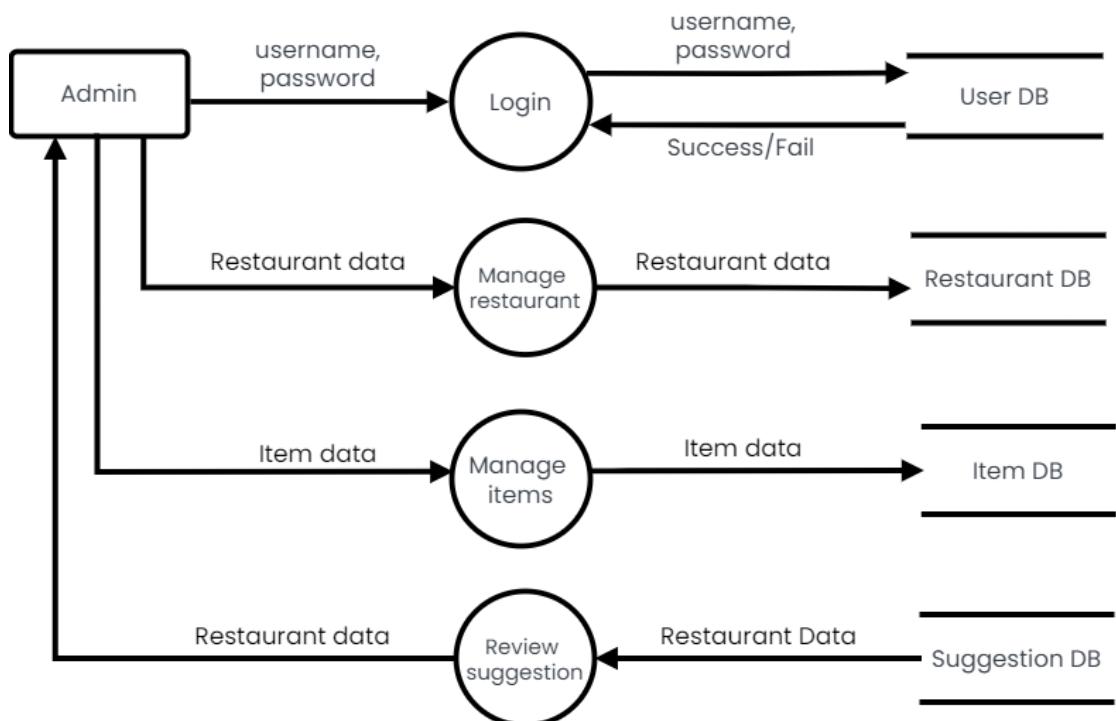
DIAGRAM	DESCRIPTION
	Represents source or destination of data
	Represents a process that transforms incoming data into outgoing flows
	Represents data flow
	Represents data stores

A Data Flow Diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where it goes and how it gets stored. DFD provide critical insights into the systems and way the information passes through it. DFD helps structure every element of the system to keep them logically intact and interconnected. On the other hand, you have the customers who need to know what is going on in a digestible easy to follow manner.

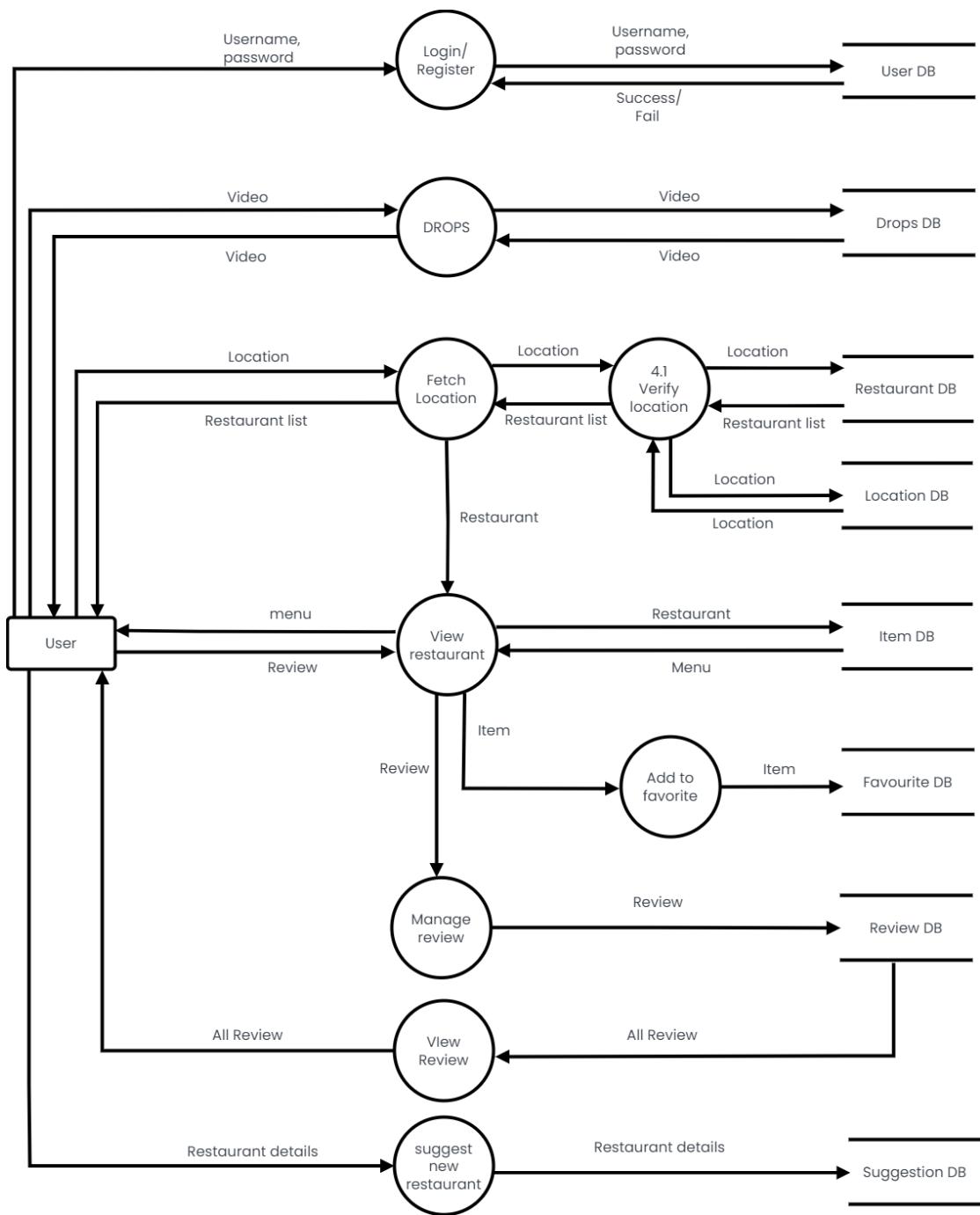
### 2.6.1.LEVEL 0 DFD OR (CONTEXT FLOW DIAGRAM)



### 2.6.2.LEVEL 1 DFD (ADMIN)



### 2.6.3.LEVEL 1 DFD (USER)



### **3 SYSTEM MODELING**

### 3.1 CLASS DIAGRAM

A Class diagram in the Unified Modeling Language is a type of static structure that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among the objects. The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

A class notation consists of three parts:

- Class Name: The name of the class appears in the first partition.
- Class Attributes: Attributes are shown in the second partition.

The attribute type is shown after the colon. Attributes map onto member variables (data members) in code.

- Class Operations (Methods): Operations are shown in the third partition.

#### Visibility

To specify the visibility of a class member (i.e. any attribute or method), these notations must be placed before the member's name:

- + Public
- Private
- # Protected
- ~ Package

#### Class Relationships

A class may be involved in one or more relationships with other classes.

##### Inheritance (or Generalization):

- Represents an "is-a" relationship.
- An abstract class name is shown in italics.
- SubClass1 and SubClass2 are specializations of Super Class.
- A solid line with a hollow arrowhead that points from the child to the parent class.

**Aggregation:**

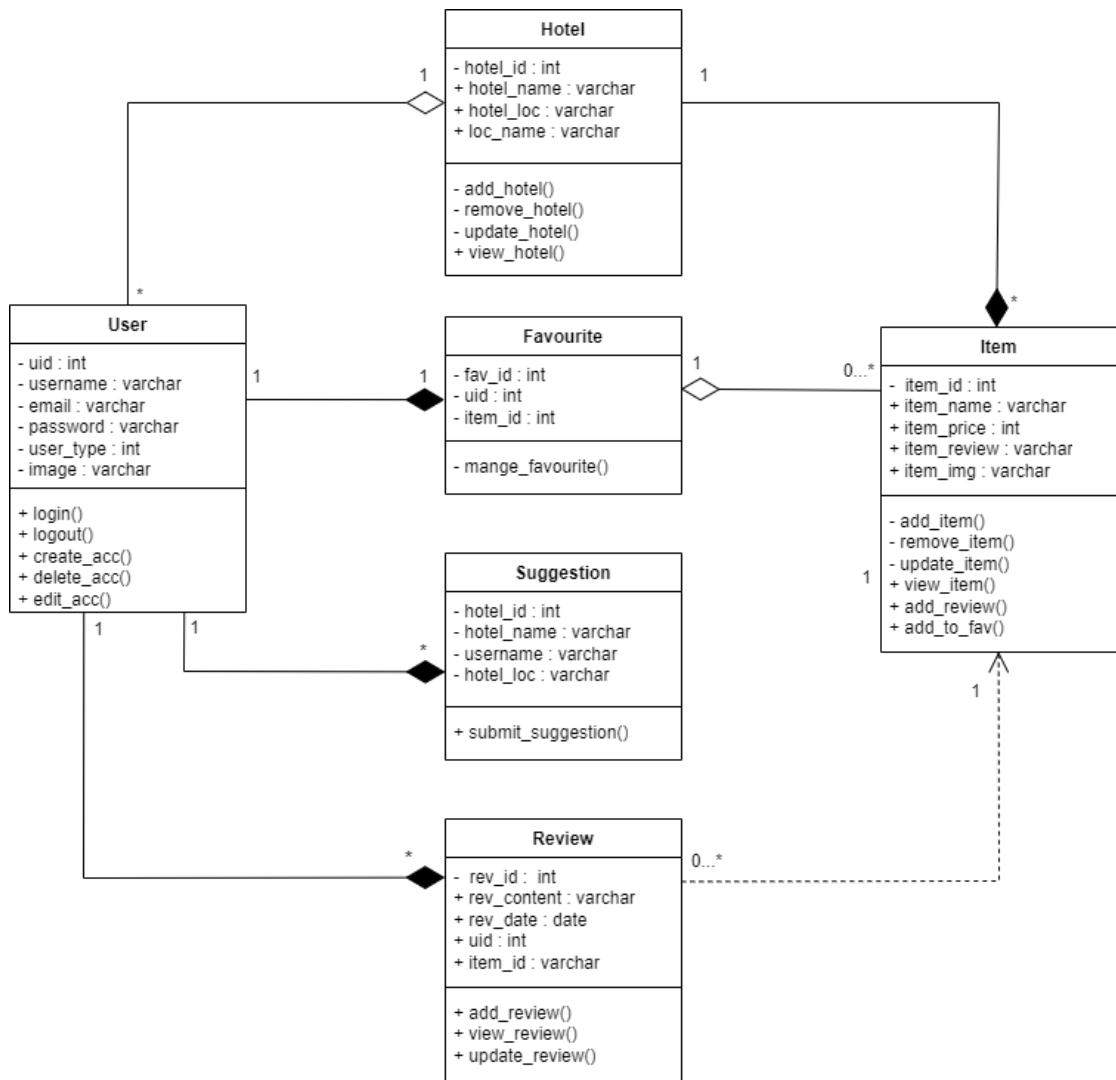
- A special type of association. It represents a "part of" relationship.
- Class2 is part of Class1.
- Many instances (denoted by the \*) of Class2 can be associated with Class1.
- Objects of Class1 and Class2 have separate lifetimes.
- A solid line with an unfilled diamond at the association end connected to the class of composite.

**Composition:**

- A special type of aggregation where parts are destroyed when the whole is destroyed.
- 
- Objects of Class2 live and die with Class1.
- 
- Class2 cannot stand by itself.
- 
- A solid line with a filled diamond at the association connected to the class of composite

**Dependency:**

- Exists between two classes if the changes to the definition of one may cause changes to the other (but not the other way around).
- Class1 depends on Class2
- 
- A dashed line with an open arrow



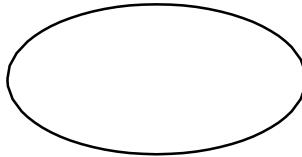
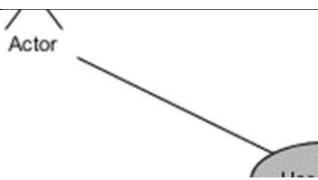
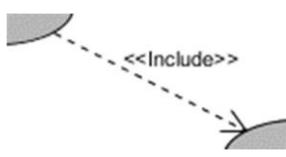
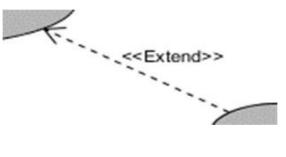
(Fig 3.1)

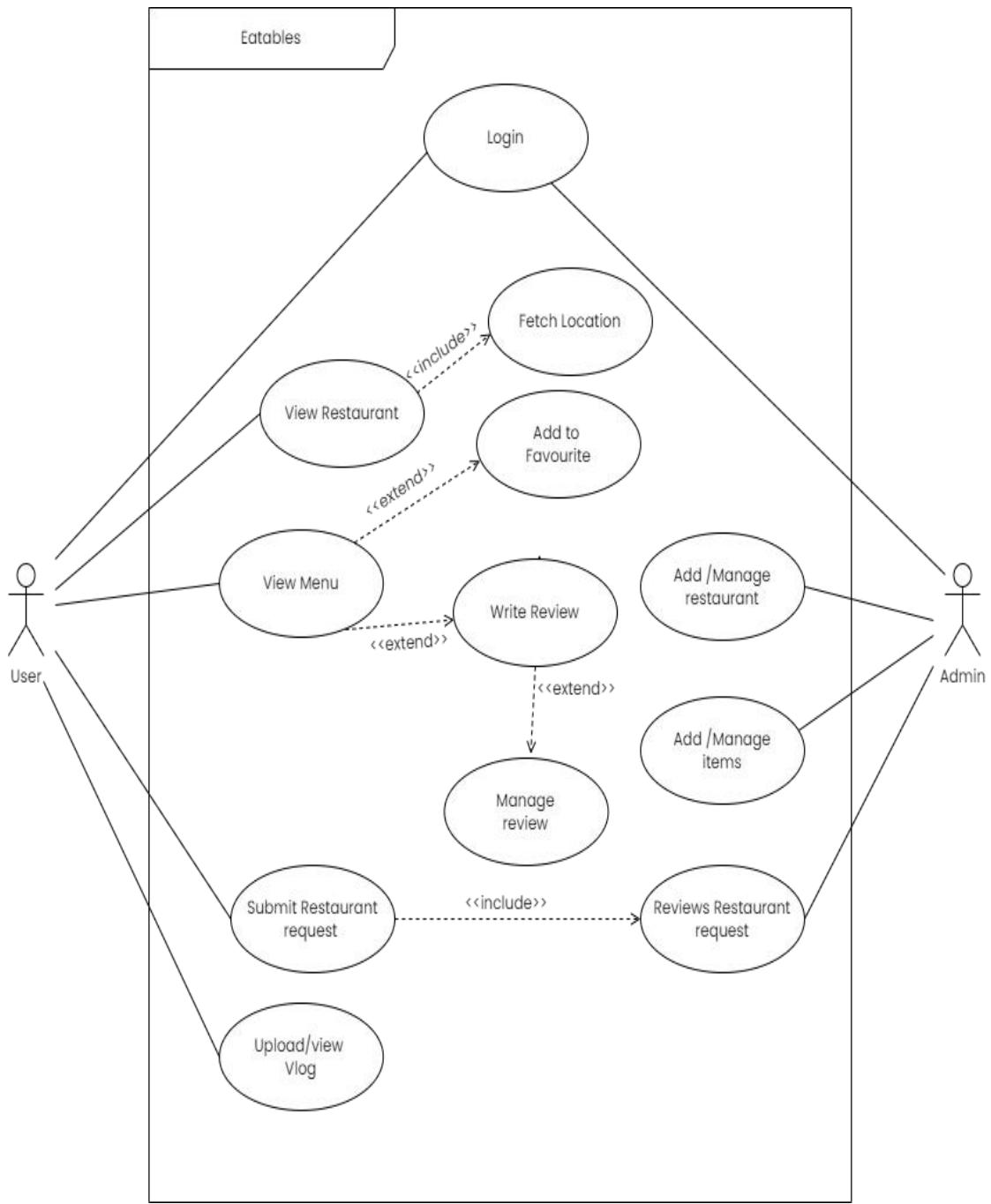
### **3.2 USECASE**

A use case diagram is a graphic depiction of the interactions among the elements of a system. It is a methodology used in system analysis to identify, clarify and organize system requirements. A use case diagram does not show the detail of the use case, it only summarizes some of the relationships between use case and the actors. In particular, the diagram does not show the order in which steps are performed to achieve the goals of each use case. It deals only with the functional requirement of the system. Use case diagrams are usually referred to as behaviour diagrams, which is used to describe set of actions that systems should or can perform in collaboration with one or more external users of the system ie Actors. Each use case should provide one or more observable and valuable result to the actor.

The purpose of use case diagram is to capture core functionalities of a system and visualize the interactions of various things called as actors with the use case. A use case consists of use cases, persons, or various things that are invoking the features called as actors and the elements that are responsible for implementing the use cases. Use case diagrams capture the dynamic behaviour of live system. Use case diagrams are responsible for visualizing the external things that interact with part of the system.

A use case diagram should be simple as possible. A use case diagram should be complete. A use case diagram should represent all interactions with use case. If there are too many use cases or actors, only the essential use cases should be represented. A use case diagram should describe at least a single module of a system. If the use case diagram is large then it should be generalized. The name of the actor and use case should be meaningful and relevant to the system. Interaction of an actor with the use case must be defined clearly and in an understandable way. Annotation must be used where ever they are required. If a use case or an actor has multiple relationship, then only significant interactions must be displayed.

DIAGRAM	DESCRIPTION
 Use Case	<p>A use case represents a user goal that can be achieved by accessing the system or software application. In Visual Paradigm, you can make use of the sub-diagram feature to describe the interaction between user and system within a use case by creating a sub-sequence diagram under a use case.</p>
 Association	<p>Actor and use case can be associated to indicate that the actor participates in that use case. Therefore, an association correspond to a sequence of actions between the actor and use case in achieving the use case.</p>
 Actor	<p>Actors are the entities that interact with a system. Although in most cases, actors are used to represent the users of system, actors can actually be anything that needs to exchange information with the system. So, an actor may be people, computer hardware, other systems, etc.</p>
 Include	<p>An include relationship specifies how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.</p>
 Extend	<p>An extend relationship specifies how the behavior of the extension use case can be inserted into the behavior defined for the base use case.</p>



(Fig 3.2)

### **3.3 SEQUENCE DIAGRAM**

UML sequence diagram model is the flow of logic within the system in a visual manner, enabling both to document and validate the logic and are commonly used for both analysis and design purposes. It is the most popular UML artifact for dynamic modelling which focus on identifying the behaviour within the system.

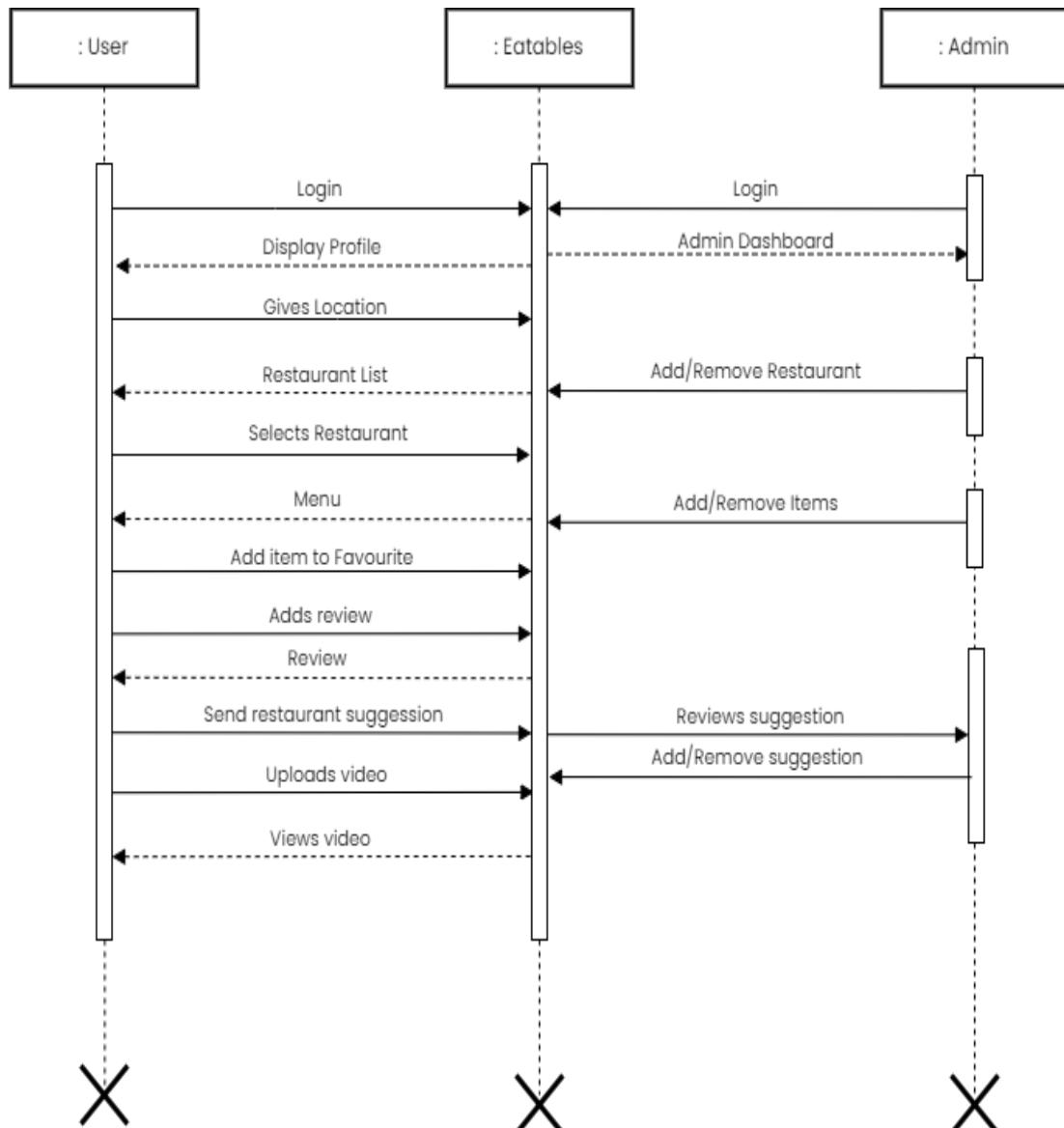
Sequence diagram are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometime called as event diagrams and event scenarios.

A sequence diagram shows, as parallel vertical lines different processes or the objects that live simultaneously and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

A sequence diagram shows object interactions arranged in time sequence, it depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

DIAGRAM	DESCRIPTION
 Object symbol	<p>Represents a class or object in UML.</p> <p>The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape.</p>
 Activation box	<p>Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.</p>
 Lifeline symbol	<p>Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labelled rectangle shape or an actor symbol.</p>

 Synchronous message symbol	<p>Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.</p>
 Asynchronous message symbol	<p>Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.</p>
 Reply message symbol	<p>Represented by a dashed line with a lined arrowhead, these messages are replies to calls.</p>
 Delete message symbol	<p>Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.</p>

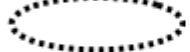
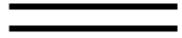


(Fig 3.3)

## **4 DATABASE MODELING**

## 4.1 ER DIAGRAM

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Here are the geometric shapes and their meaning in an E-R Diagram.

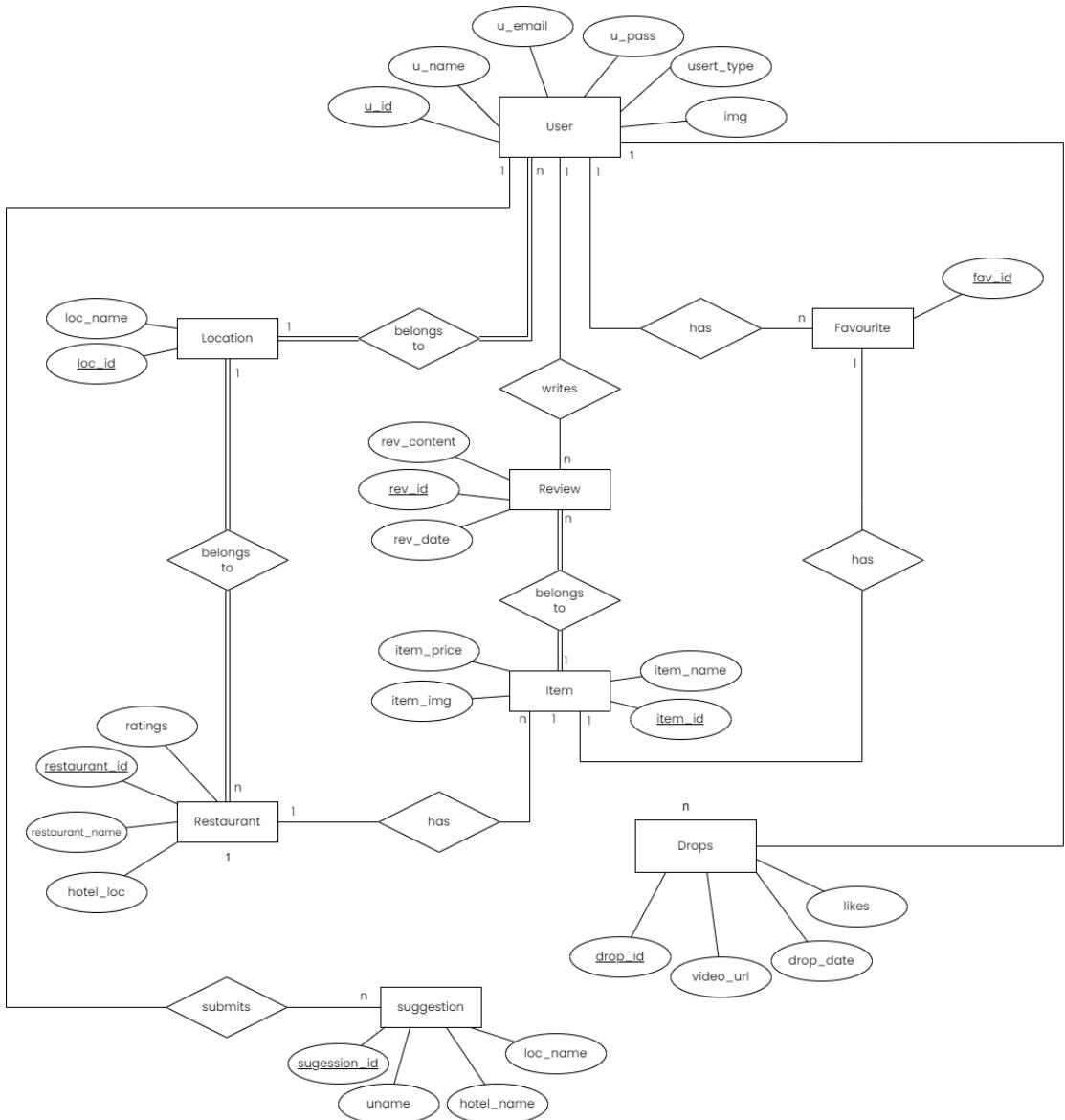
	Represents Entity
	Represents Attribute
	Represents Relationship
	Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)
	Represents Multivalued Attributes
	Represents Derived Attributes
	Represents Total Participation of Entity
	Represents Weak Entity
	Represents Weak Relationships
	Represents Composite Attributes
	Represents Key Attributes / Single Valued Attributes

ER diagram has three main components:

- **Entity:** An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.
- **Attribute:** An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:
  - **Key attribute:** A key attribute can uniquely identify an entity from an entity set. Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.
  - **Composite attribute:** An attribute that is a combination of other attributes is known as composite attribute.
  - **Multivalued attribute:** An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in an ER Diagram.
  - **Derived attribute:** A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by dashed oval in an ER Diagram.
- **Relationship:** A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities.

There are four types of relationships:

- **One to One Relationship:** When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.
- **One to Many Relationship:** When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.
- **Many to One Relationship:** When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship.
- **Many to Many Relationship:** When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.



(Fig 4.1)

## 4.2 TABLE DESCRIPTION

A table is an arrangement of data in rows and columns, or possibly in a more complex structure. A table is a collection of related data held in a table format within a database. The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyse the data.

A database consists of one or more tables. Each table is made up of rows and columns. Each row in a relational table is uniquely identified by a primary key. This can be by one or more sets of column values. In most scenarios it is a single column, such as student ID.

Every relational table has one primary key. Its purpose is to uniquely identify each row in the database. No two rows can have the same primary key value. The practical result of this is that you can select every single row by just knowing its primary key.

### 4.2.1. Normalization

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updating anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

Normalization is the process of organizing the data in the database.

Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.

Normalization divides the larger table into smaller tables and links them using relationship.

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2 NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transitive dependency exists.

- **First Normal Form(1NF)**

A relation will be 1NF if it contains an atomic value.

It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.

First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

- **Second Normal Form(2NF)**

In the 2NF, relational must be in 1NF.

In the second normal form, all non-key attributes are fully functional dependent on the primary key

- **Third Normal Form(3NF)**

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

3NF is used to reduce the data duplication. It is also used to achieve the data integrity.

If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency  $X \rightarrow Y$ .

$X$  is a super key.

$Y$  is a prime attribute, i.e., each element of  $Y$  is part of some candidate key.

USER			
Column name	Data types	Constraints	Description
UID	INT	PRIMARY KEY	User ID
UNAME	VARCHAR(20)	NOT NULL	User name
USER_TYPE	VARCHAR(8)	NOT NULL	User type consists either Admin or customer
FULLNAME	VARCHAR(30)	NOT NULL	User full name
EMAIL	VARCHAR(30)	NOT NULL	User email address for login
PASSWORD	VARCHAR(16)	NOT NULL	User Password for login
RESET_TO-KEN	VARCHAR(100)	NULL	Token used for resetting password
RESET_EXPIRATION	VARCHAR(30)	NULL	Expiry date of reset token
IMG	VARCHAR(100)	NULL	User profile image
VERIFIED	INT	NULL	Identify verified users

HOTEL			
Column name	Data types	Constraints	Description
HOTEL_ID	INT	PRIMARY KEY	This field is used to uniquely identify hotels
HO-TEL_NAME	VARCHAR(30)	NOT NULL	Name of the hotel
HOTEL_LOC	VARCHAR(100)	NULL	Latitude and longitude of hotel location
LOC_NAME	VARCHAR(30)	NOT NULL	Location name
RATINGS	VARCHAR(10)	NULL	Ratings of the hotel
LINKS	VARCHAR(300)	NULL	External link for the hotel
DESC	VARCHAR(150)	NULL	Description for the hotel

DROPS			
Column name	Data types	Constraints	Description
DROP_ID	INT	PRIMARY KEY	To uniquely identify video uploaded to drops
UID	INT	FOREIGN KEY	Unique ID which is used to identify user
VIDEO_URL	VARCHAR(150)	NOT NULL	URL of the video file
HOTEL_NAME	VARCHAR(100)	NOT NULL	Name of the hotel
DROP_DATE	VARCHAR(20)	NULL	Date of uploading the video
LIKES	INT	NULL	Likes of the drop

FAVORITE			
Column name	Data types	Constraints	Description
FAV_ID	INT	PRIMARY KEY	To uniquely identify favourite list of each users
UID	INT	FOREIGN KEY	Unique ID for users
ITEM_ID	INT	FOREIGN KEY	Unique ID for items

REVIEW			
Column name	Data types	Constraints	Description
REVIEW_ID	INT	PRIMARY KEY	To uniquely identify reviews of each users
UID	INT	FOREIGN KEY	Unique ID of user
ITEM_ID	INT	FOREIGN KEY, NULL	Unique ID for items
REVIEW_CONTENT	VAR-CHAR(150)	NOT NULL	Review of the user
REVIEW_DATE	DATETIME	NOT NULL	Date of posting the review

SUGGESTION			
Column name	Data types	Constraints	Description
HOTEL_ID	INT	PRIMARY KEY	To uniquely identify query of each users
UNAME	VARCHAR(30)	NOT NULL	User name of user
HO- TEL_NAME	VARCHAR(30)	NOT NULL	Hotel name
HOTEL_LOC	VARCHAR(100)	NULL	Latitude and longitude
LOC_NAME	VARCHAR(30)	NOT NULL	Location name
RATINGS	VARCHAR(10)	NULL	Ratings of the hotel
LINKS	VARCHAR(300)	NULL	External link for the hotel
DESC	VARCHAR(150)	NULL	Description of the hotel

LOCATION			
Column name	Data types	Constraints	Description
LOC_ID	INT	NOT NULL	Unique ID for each location
LOC_NAME	VARCHAR(30)	PRIMARY KEY	Location name

ITEM			
Column name	Data types	Constraints	Description
ITEM_ID	INT	PRIMARY KEY	Unique ID for items
HOTEL_ID	INT	FOREIGN KEY	Unique ID for hotels
ITEM_NAME	VARCHAR(20)	NOT NULL	Name of the item
ITEM_PRICE	INT (4)	NOT NULL	Price of the item
ITEM_RAT- ING	INT	NULL	Rating for each item
ITEM_IMG	VARCHAR(300)	NULL	Rating given for the item

## **5. TESTING AND VALIDATION**

## **5.1.INTRODUCTION**

Software development project, errors can be introduced at any stage during development. Testing the capabilities of a system is a very important task. Users interested in getting their job done and test cases should be chosen to identify the system that will stop them from doing their job. Although errors such as seen corruption are irritating, they are less disruptive than errors. It's important that the system works under normal usage conditions than under occasional conditions that only arise with extreme data values.

## **5.2.TESTING STEPS**

### **5.2.1.Unit Testing**

Unit tests are written from a programmer's perspective. They ensure that a particular method of a class successfully performs a set of specific tasks. Each test confirms that a method produces the expected output when given a known input. A series of stand-alone tests are conducted during unit testing. Each test examines an individual component that is new or has been modified. Unit tests focus on functionality and reliability. Unit testing is done in a test environment prior to the system integration. If a defect is discovered during a unit test, the severity of the defect will dictate whether or not it will be fixed before the module is approved.

### **5.2.2.Integration Testing**

Integration testing examines all the components and modules that are new, changed, affected by the change, or needed to form a complete system. Integration testing requires the involvement of other systems and interface with other API s, s, external partners, etc.

It is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and proceeds with system testing. Integration testing takes in its input modules that have been tested, groups them in large aggregates, applies tests defined in an integration test plan without aggregates, and delivers as its output the integrated

system ready for system testing.

### **5.2.3.Validation**

Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in the manner that is expected by the client. After validation test has been conducted, one of the three possible conditions exists. The function or performance characteristics confirm to specifications and are accepted. The deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation test and found it to be working satisfactorily.

### **5.2.4.Output testing**

After performing the validation, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also, the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system

### **5.2.5.User acceptance testing**

Acceptance testing is the level of software testing where a system is tested for acceptability. It is a formal testing with respect to other needs, requirements conducted to determine whether or not a system satisfies an acceptance criterion and to enable the users or others unauthorized to determine whether or not to accept a system.

## 5.3.TEST CASES AND VALIDATION

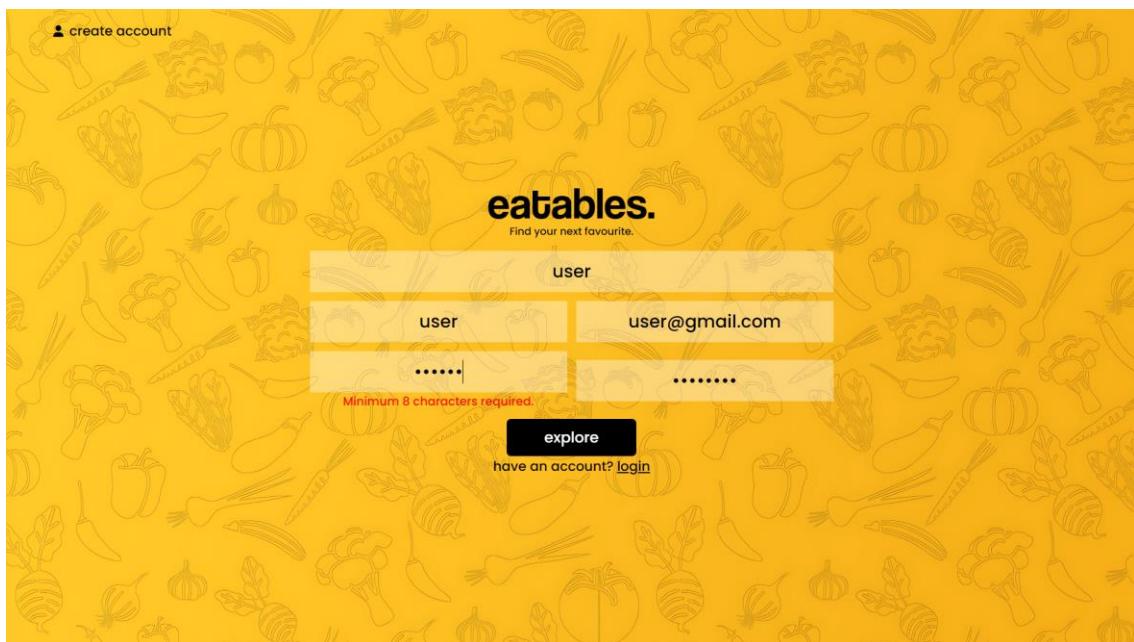
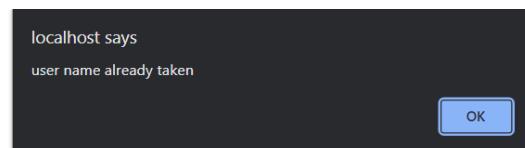
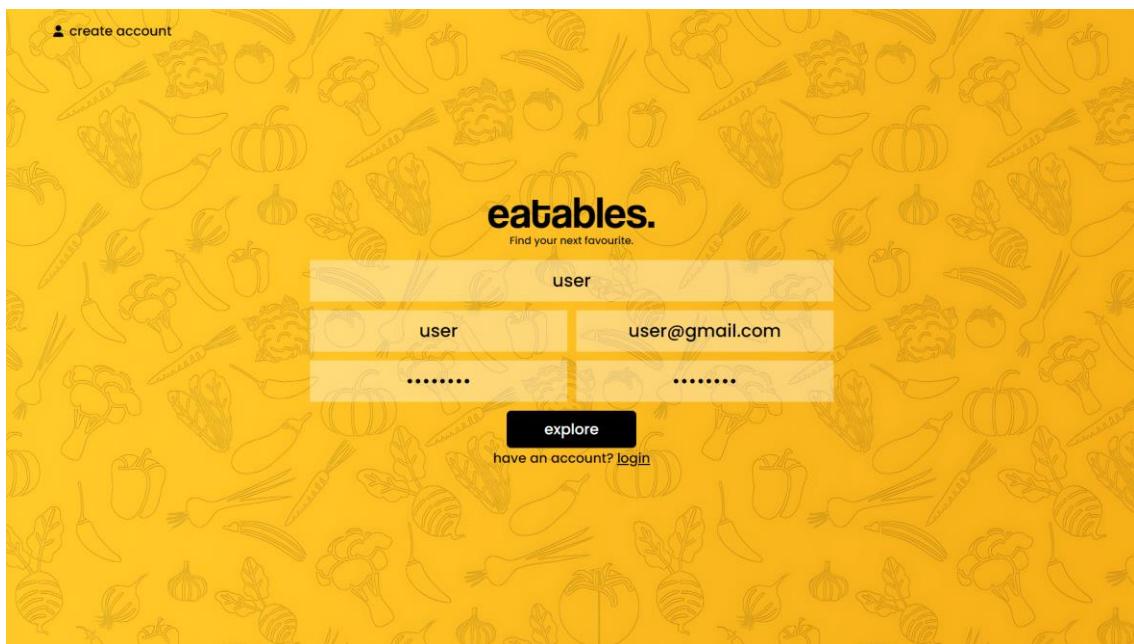
### 5.3.1.Sign up form

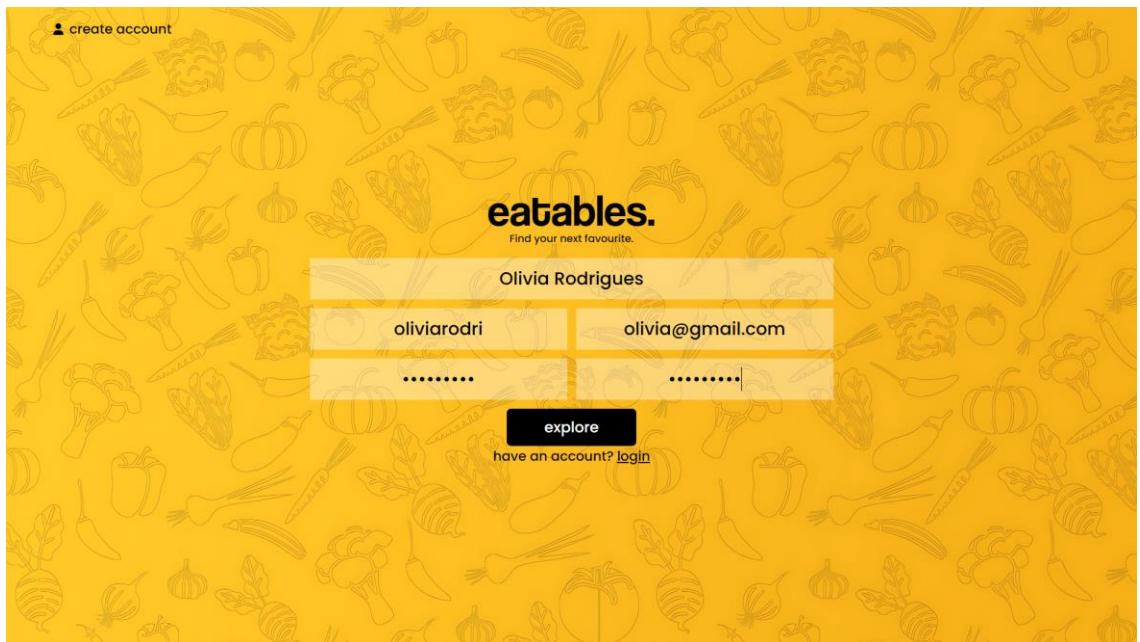
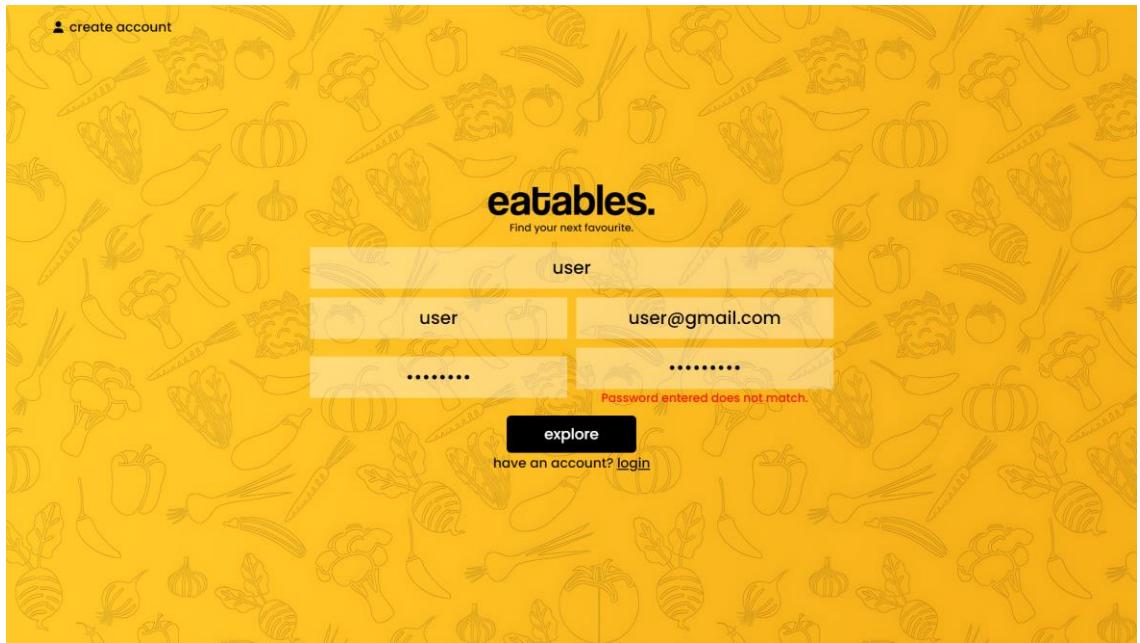
Test Case ID	Test Condition	Expected Output	Result
1	If all field empty	Fill all the credentials	Success
2	If full name field empty	Please enter your full name	Success
3	If username field empty	Please enter your user name	Success
4	If email field empty	Please enter your email address	Success
5	If password field empty	Please enter a password	Success
6	If password confirm field empty	Please confirm your password	Success
7	If full name field consist number/special characters	Only alphabets and whitespaces are allowed	Success
8	If username field consist of less than 4 or greater than 15 characters	Username requires 4 – 15 characters	Success
9	If user name field consist number/special characters/whitespace/uppercase	Only lowercase letters are allowed	Success
10	If user name field consist of existing username	Username already taken	
11	If email field consist of invalid email	Invalid email format	Success
12	If email field consisting of existing email	Account already exist for this email	Success

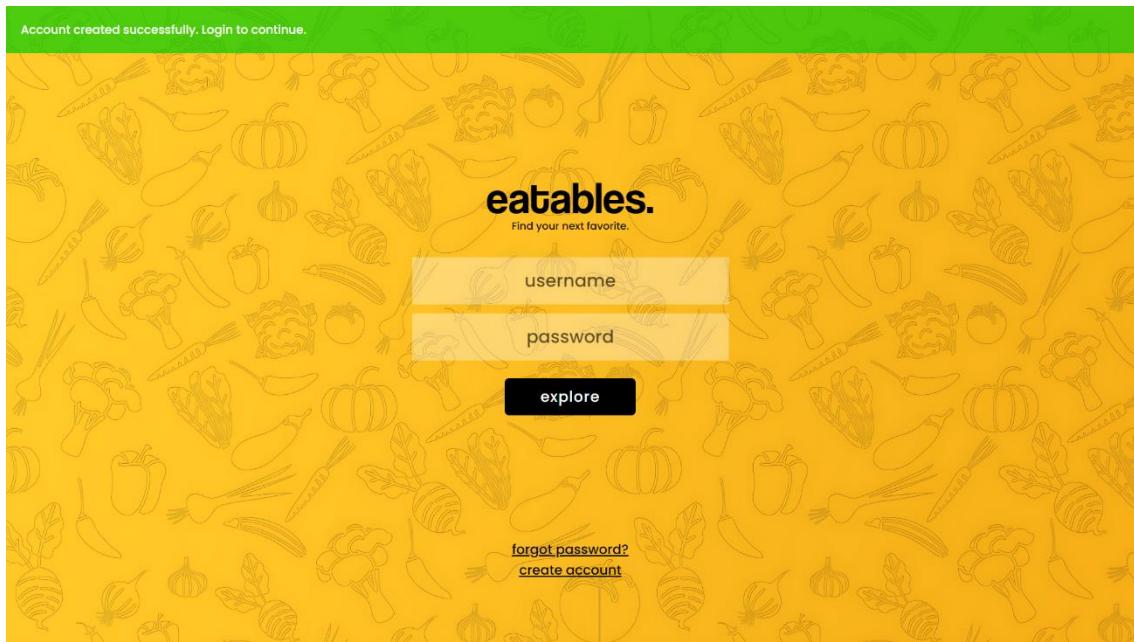
13	If password field consist of less than 8 characters	Minimum 8 characters required	Success
14	If password field does not contain combination of number and letter	Combination of alphabets and numbers required	Success
15	If confirm password field does not match password field	Password entered does not match	Success
16	If all credentials are valid	Account created successfully and redirect to login page.	Success

### Sign up page:

The screenshot shows the sign-up page for the eatables platform. The page has a yellow background with a repeating pattern of various vegetables like tomatoes, carrots, and bell peppers. At the top left, there's a small 'create account' button. In the center, the 'eatables' logo is displayed with the tagline 'Find your next favourite.' Below the logo is a sign-up form. The form consists of five input fields arranged in a grid: 'fullname' (top row), 'username' and 'email' (second row), 'password' and 'confirm-password' (third row). To the right of the 'password' field, there's a note: 'Fill all credentials! have an account? [login](#)'. At the bottom left of the form area, there's a red 'explore' button.

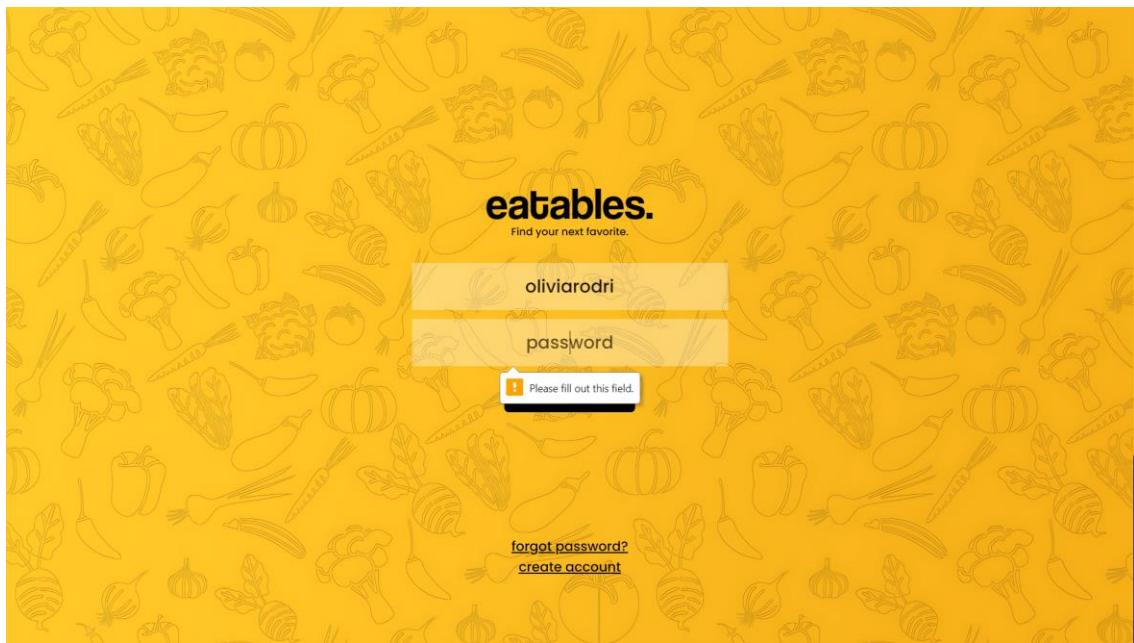
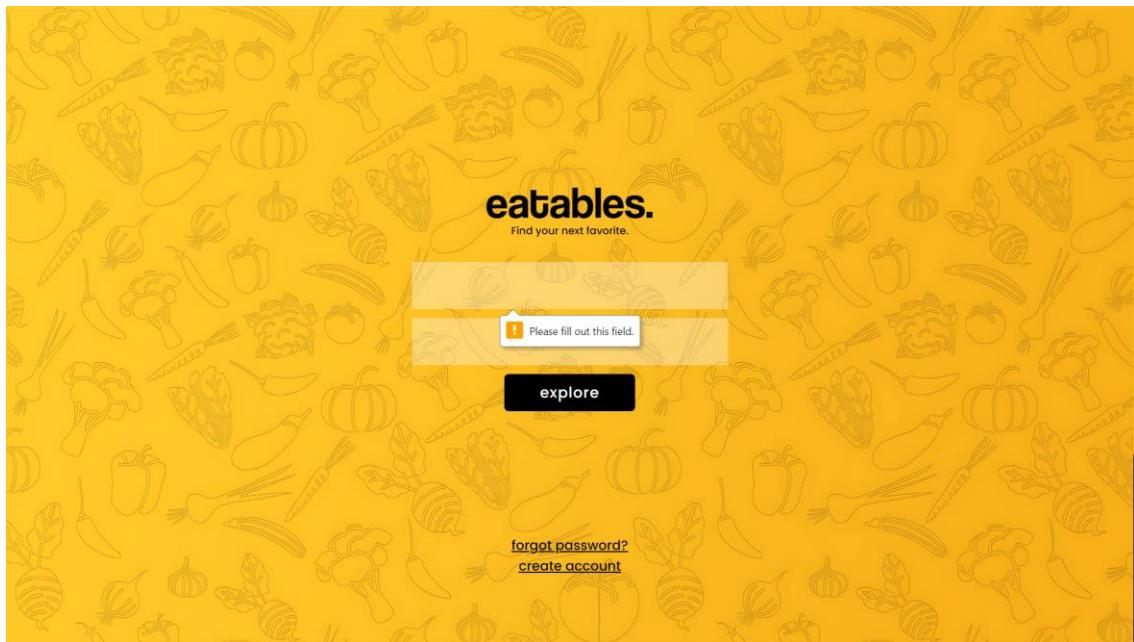


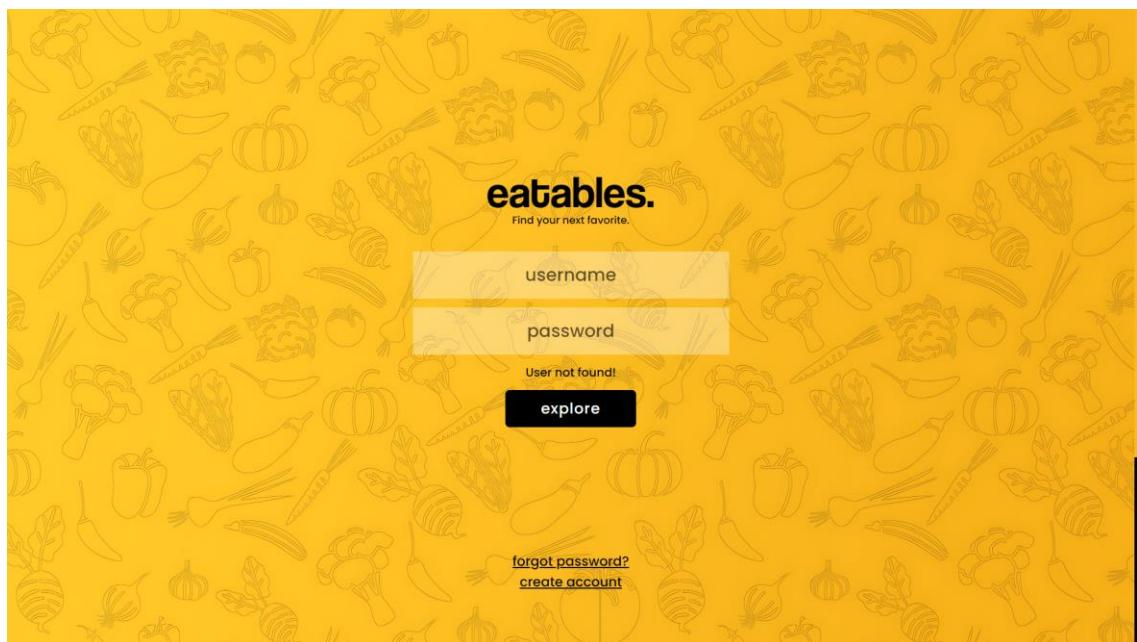
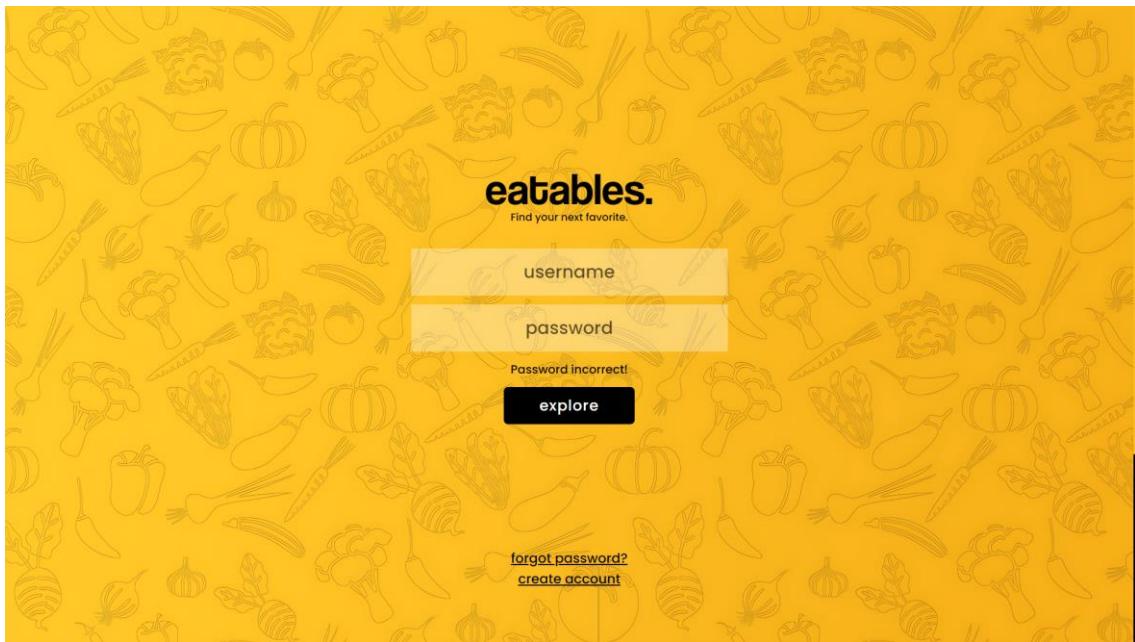




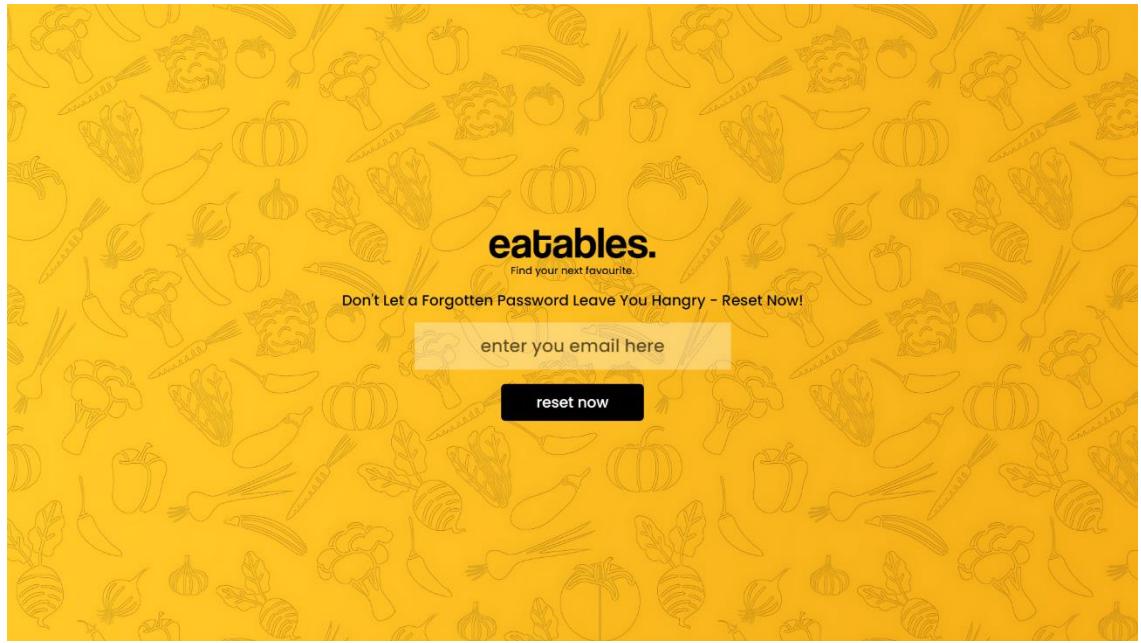
### 5.3.2.Login form

<b>Test Case ID</b>	<b>Test Condition</b>	<b>Expected Output</b>	<b>Result</b>
1	If username field empty	Please fill out this field	Success
2	If password field empty	Please fill out this field	Success
3	If inputs wrong username	User not found	Success
4	If inputs wrong password	Password incorrect	Success
5	If user clicks on forgot password	Redirect to reset password page	Success
6	If username and password are valid for user	Redirect to user home page	Success
7	If username and password are valid for admin	Redirect to admin profile page	Success

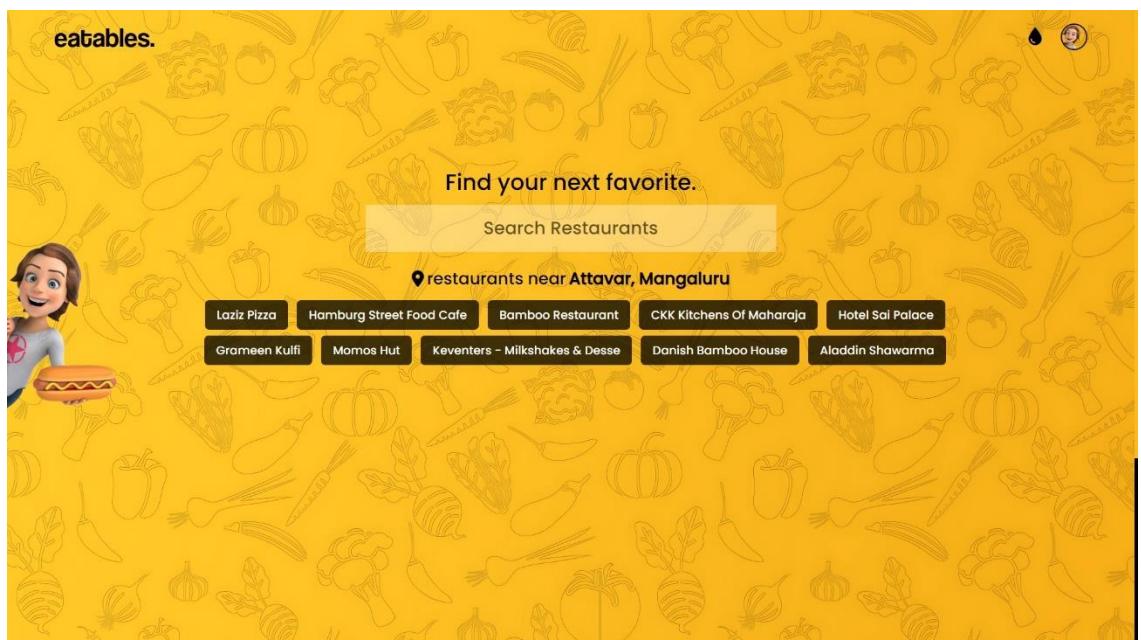




Login page>>Forgot password:

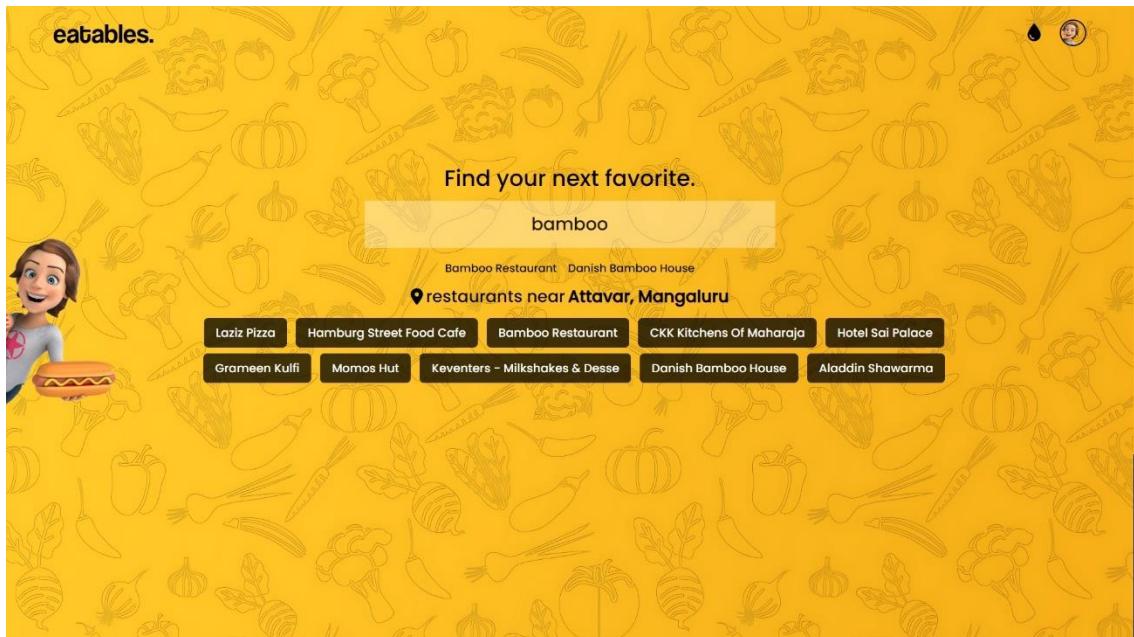


login page>>home page (login successful):



### 5.3.3.Home page

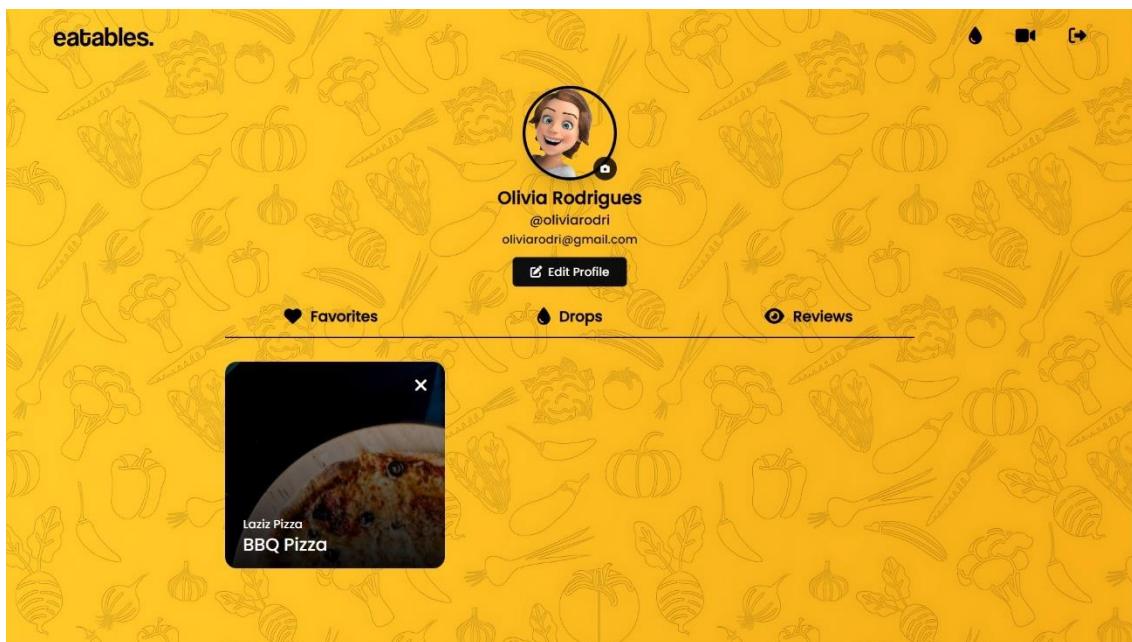
Test Case ID	Test Condition	Expected Output	Result
1	If user types restaurant name in search box.	Display similar restaurant names under search box	Success
2	If user clicks on eatables logo.	Page refreshes	Success
3	If user clicks on Drops icon	Redirect to Drops page	Success
4	If user clicks on profile icon	Redirect to user profile page	Success
5	If user chooses a restaurant name	Redirect to item details page	Success
6	If user click on assistant	Open filter window	Success
7	If user applies price filter	Display items based on the price	Success
8	If user clicks on Submit restaurant in footer	Redirect to restaurant suggestion page	Success



Home page>>Drops:



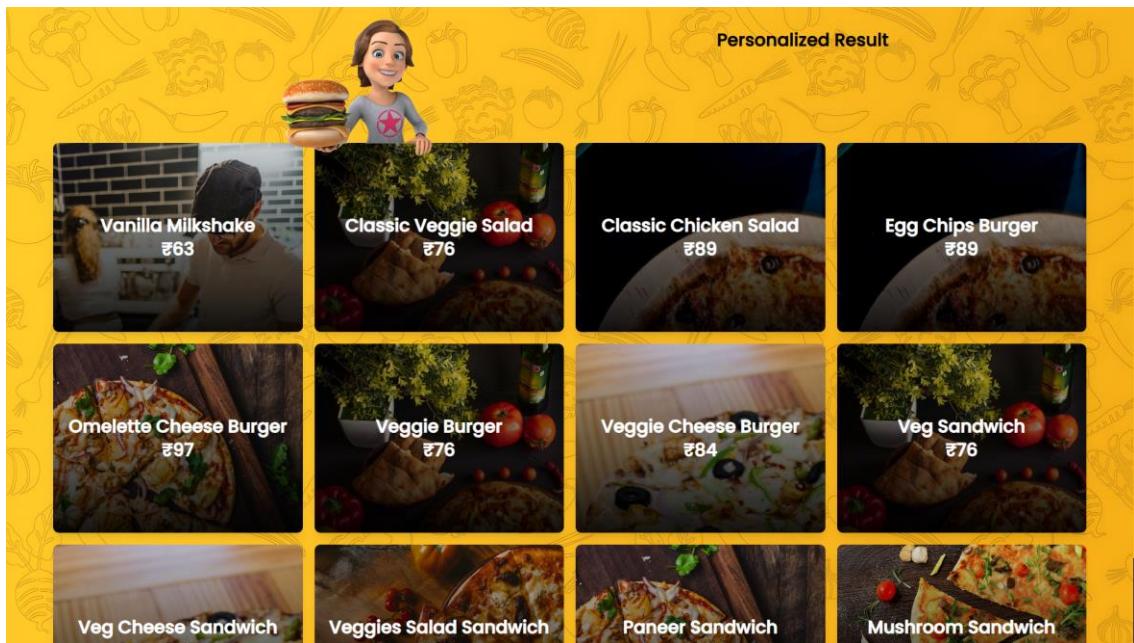
Home page>>user profile:



Home page>>Hotel:

The screenshot shows a restaurant profile for "Laziz Pizza". The top left features the eatables logo and the restaurant's name. Below it is a contact section with a phone number (9232387927) and an email address (youreatable@gmail.com). A 5-star rating icon is present. The restaurant's description is a placeholder text: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, blanditiis ratione modi consectetur veritatis temporibus repellendus! Quas amet sit sunt eos neque eum tenetur, maxime possimus dolorum deleniti delectus doloribus." A "zomato" badge is visible. To the right is a map showing the location of the restaurant. Below the map is a grid of eight pizza images with their names: "BBQ Pizza", "Special Spicy Pizza", "Lions Appetite Pizza", "Paneer Ghee Roast Pizza", "Cheese N Cheese Pizza", "Pizza Box for One [Veg]", "Pizza Box for One [Non-Veg]", and "Family pizza box [Veg]".

The screenshot shows the eatables homepage. The top features the eatables logo and a search bar with the placeholder "Find your next favorite.". Below the search bar is a section titled "restaurants near Attavar, Mangaluru" with a list of nearby restaurants: Laziz Pizza, Hamburg Street Food Cafe, Bamboo Restaurant, CKK Kitchens Of Maharaja, Hotel Sai Palace, Grameen Kulfi, Momos Hut, Keventers - Milkshakes & Desses, Danish Bamboo House, and Aladdin Shawarma. On the left side, there is a cartoon character named Sera holding a hotdog, and a sidebar with filtering options: "Want to have filtered result?", "yes" (button), "price" (button), and "Low" / "Medium" / "High" (radio buttons).



▼ RESTAURANTS NEAR ATTAVVAI, MANGALURU


[Laziz Pizza](#)
[Hamburg Street Food Cafe](#)
[Bamboo Restaurant](#)
[CKK Kitchens Of Maharaja](#)
[Hotel Sai Palace](#)

[Grameen Kulfi](#)
[Momos Hut](#)
[Keventers - Milkshakes & Desse](#)
[Danish Bamboo House](#)
[Aladdin Shawarma](#)

Add a missing restaurant. Submit restaurant

**eatables.**

Unlock the flavors of the world with Eatables:  
Your ultimate food database.

[Twitter](#) [Instagram](#) [YouTube](#)

© 2023-24 eatables Inc.  
localhost/eatables/add-restaurant.php

Home page>>Submit restaurant:

Found a restaurant that's not on our list?  
Fill the details.

RESTAURANT NAME\*: Hotel Chicken Hut

LOCATION\*: Manglore

LOCATION (GMAP LOCATION LINK)\*: <https://www.gmap.com/z09io1coiws1q>

RATING(1-5)\*: 4

LINK (ZOMATO LINK): <https://www.zomato.com/z09io1coiws1q>

DESCRIPTION:

Add

\*required

### 5.3.4. Submit restaurant

TestCase ID	Test Condition	Expected Output	Result
1	If restaurant name field empty	Display please fill out this field	Success
2	If location field empty	Display please fill out this field	Success
3	If ratings field empty	Display please fill out this field	Success
4	If link field empty	Display please fill out this field	Success
5	If description field empty	Display please fill out this field	Success
6	If all fields are valid	Display submit message and redirect to user profile page	Success

**eatables.**

Found a restaurant that's not on our list?  
Fill the details.

RESTAURANT NAME\*: Manglore

LOCATION: Manglore

LOCATION (GMAP LOCATION LINK)\*: <https://www.gmap.com/haiuhgiw94>

RATING(1-5)\*: 4

LINK (ZOMATO LINK): <https://www.zomato.com/haiuhgiw94>

DESCRIPTION: Hotel in kankanady

Add \*required

**eatables.**

Found a restaurant that's not on our list?  
Fill the details.

RESTAURANT NAME\*: Hotel Chicken Hut

LOCATION: Manglore

LOCATION (GMAP LOCATION LINK)\*: [https://www.gmap.com/haiuhgiw94](#)

RATING(1-5)\*: 4

LINK (ZOMATO LINK): <https://www.zomato.com/haiuhgiw94>

DESCRIPTION: Hotel in kankanady

Add \*required

**eatables.**

Found a restaurant that's not on our list?  
Fill the details.

RESTAURANT NAME\*: Hotel Chicken Hut

LOCATION\*: Manglore

LOCATION (GMAP LOCATION LINK)\*: <https://www.gmap.com/haiuhgiw94>

RATING(1-5)\*:

LINK (ZOMATO LINK):  Please fill out this field.  
<https://www.zomato.com/haiuhgiw94>

DESCRIPTION: Hotel in kankanady

**Add**

\*required

**eatables.**

Found a restaurant that's not on our list?  
Fill the details.

RESTAURANT NAME\*: Hotel Chicken Hut

LOCATION\*: Manglore

LOCATION (GMAP LOCATION LINK)\*: <https://www.gmap.com/ofagb24b8>

RATING(1-5)\*:  4

LINK (ZOMATO LINK):   
<https://www.zomato.com/ofagb24b8>

DESCRIPTION: Hotel in kankanady

**Add**

\*required

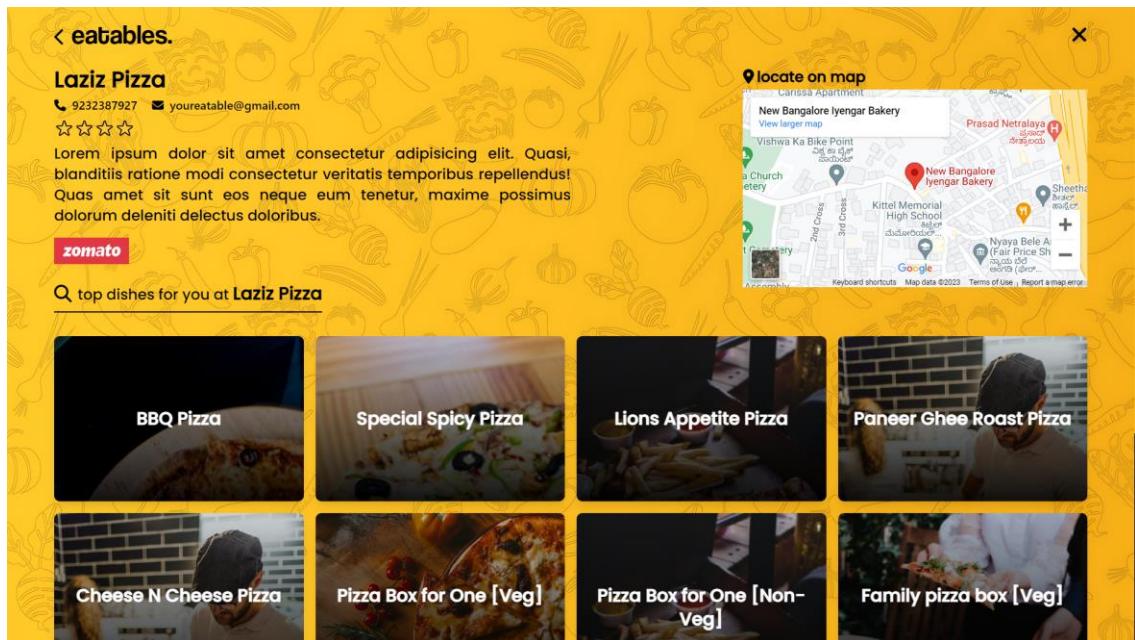
localhost says

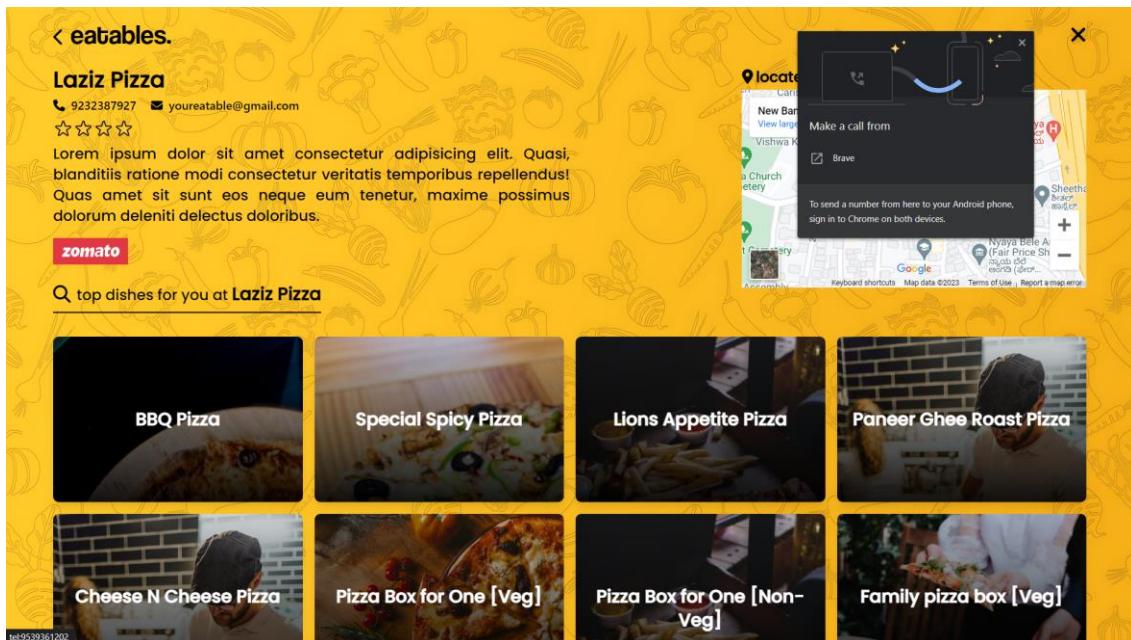
Suggestion submitted successfully.

OK

### 5.3.5.Hotel page

Test Case ID	Test Condition	Expected Output	Result
1	If user clicks on contact number	Dial the contact number	Success
2	If user clicks on email	Open gmail and compose a new mail	Success
3	If user clicks/drags on map	Show location of the restaurant	Success
4	If user clicks on zomato icon	Redirect to zomato page of the restaurant	Success
5	If user clicks on an item	Redirect to item details page	Success
6	If user clicks on number in pagination	Redirect to the page	Success
7	If user clicks on eatables	Redirect to home page	Success





**zomato** Mangalore  Search for restaurant, cuisine or a dish

Log in Sign up

Home / India / Mangalore / Attavar / Laziz Pizza / Order Online





**Laziz Pizza**

Pizza, Fast Food, Sandwich, Burger, Chinese, Beverages, Sichuan  
Attavar, Mangalore

Closed - Opens at 10:30am ⓘ

[Direction](#) [Bookmark](#) [Share](#)

Overview Order Online Reviews Photos Menu

**Recommended (25)**

- Combos (5)
- Salads (2)

**Order Online**

Currently closed for online ordering.  
Live tracking not available

veg only

0 Dining Reviews 4.0 \* 2,131 Delivery Reviews

View Gallery

Online ordering is only supported on the mobile app

Download the App zomato

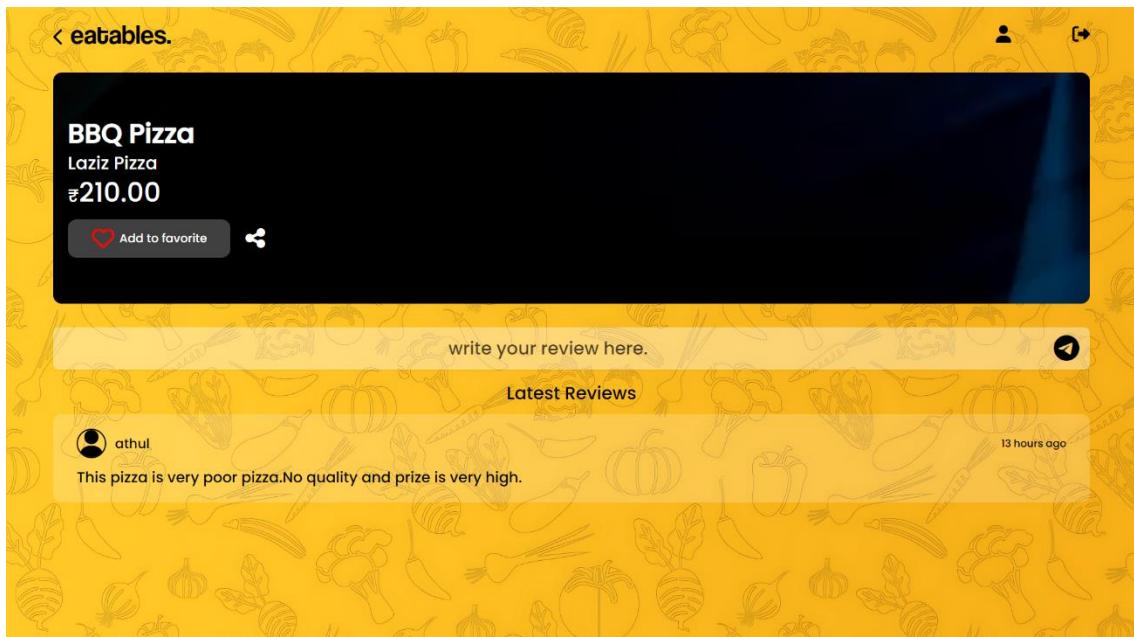
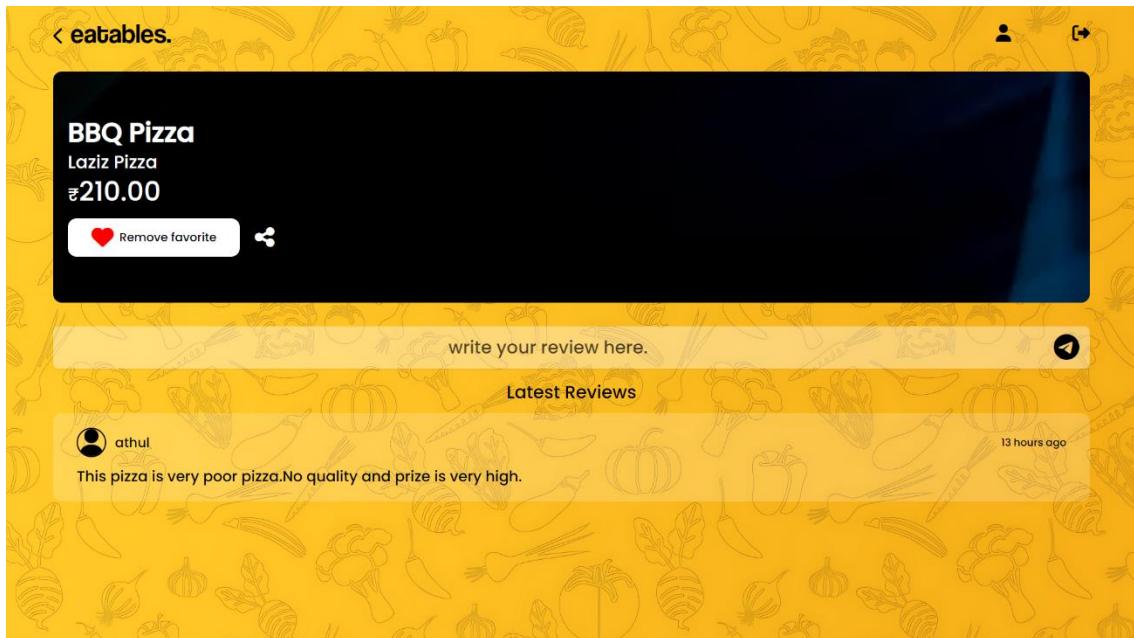
Hotel page>>item page:

The screenshot shows a mobile application interface for a food delivery service. At the top, there is a header with a back arrow and the word "eatables.". Below the header, a large black rectangular box contains the product details: "BBQ Pizza" by "Laziz Pizza" at ₹210.00. There is a "Remove favorite" button with a heart icon and a share icon. To the right of the main product box is a dark blue vertical sidebar. Below the product details, there is a yellow horizontal bar with the placeholder text "write your review here." and a back arrow icon. Underneath this bar, the heading "Latest Reviews" is displayed. A single review is shown, posted by a user named "athul" 12 hours ago. The review text reads: "This pizza is very poor pizza.No quality and prize is very high." The background of the entire screen features a repeating pattern of various vegetables like carrots, onions, and bell peppers.

The screenshot shows a mobile application interface displaying a grid of pizza options. At the top, there is a search bar with the placeholder text "Q top dishes for you at Laziz Pizza". Below the search bar, the heading "top dishes for you at Laziz Pizza" is displayed. The main content area consists of a 3x4 grid of pizza images, each with a title below it. The pizzas are: Laziz Special Combo, Margherita Pizza, Cheese N Cheese Pizza, Pomidor Pizza; Laziz Desi Veg Pizza, Corn Golden Pizza, Vegetariana Pizza, Garden Fresh Pizza; Veg Paprika Pizza, Paneer Tikka Pizza, Fiery Paneer Pizza, and Mushroom Paprika Pizza. Each pizza image is accompanied by its name. At the bottom of the grid, there is a navigation bar with five numbered buttons (1, 2, 3, 4, 5), where the second button is highlighted in yellow. The background of the entire screen features a repeating pattern of various vegetables like carrots, onions, and bell peppers.

### 5.3.6.Item page

<b>Test Case ID</b>	<b>Test Condition</b>	<b>Expected Output</b>	<b>Result</b>
1	If user clicks on eatables	Redirect to home page	Success
2	If user clicks on profile icon	Redirect to user profile page	Success
3	If user clicks on add to favourite/remove favourite	Item is added/removed from favourites accordingly	Success
4	If clicked on share icon	Website URL is copied to clipboard	Success
5	If tried to submit empty review	Display please fill out this field	Success
6	If user submits review	Store review and display it in the same page in the review section.	Success
7	If user clicks on logout icon	Log out and redirect to login page	Success



< eatables.

**BBQ Pizza**  
Laziz Pizza  
₹210.00

Add to favorite

write your review here.

Please fill out this field. vs

athul 13 hours ago

This pizza is very poor pizza.No quality and prize is very high.

< eatables.

**BBQ Pizza**  
Laziz Pizza  
₹210.00

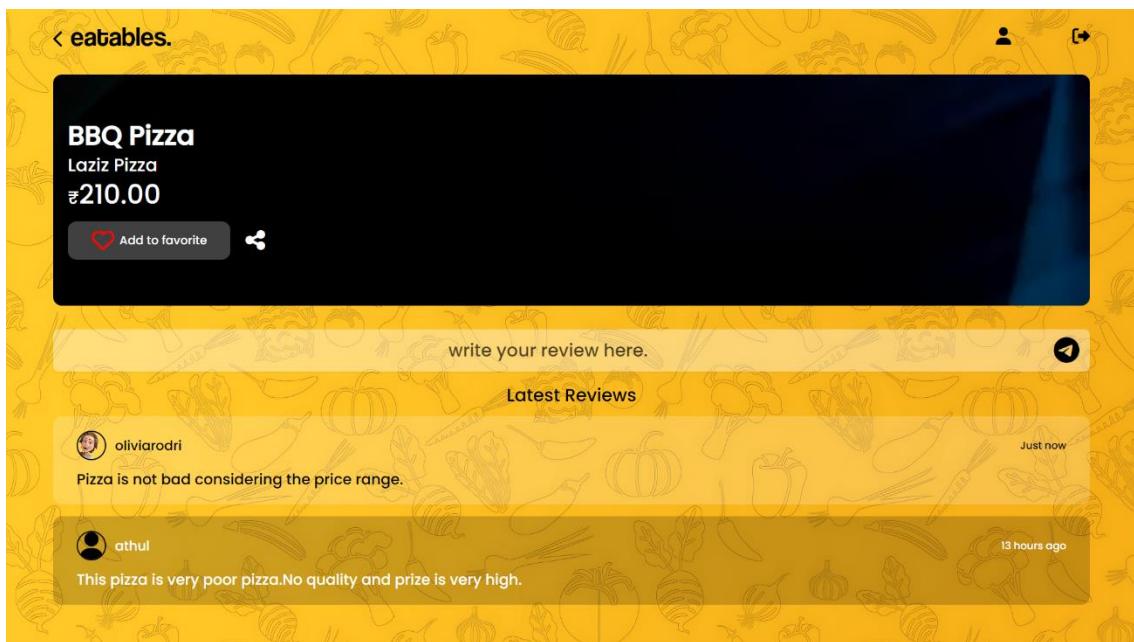
Add to favorite

Pizza is not bad considering the price range.

Latest Reviews

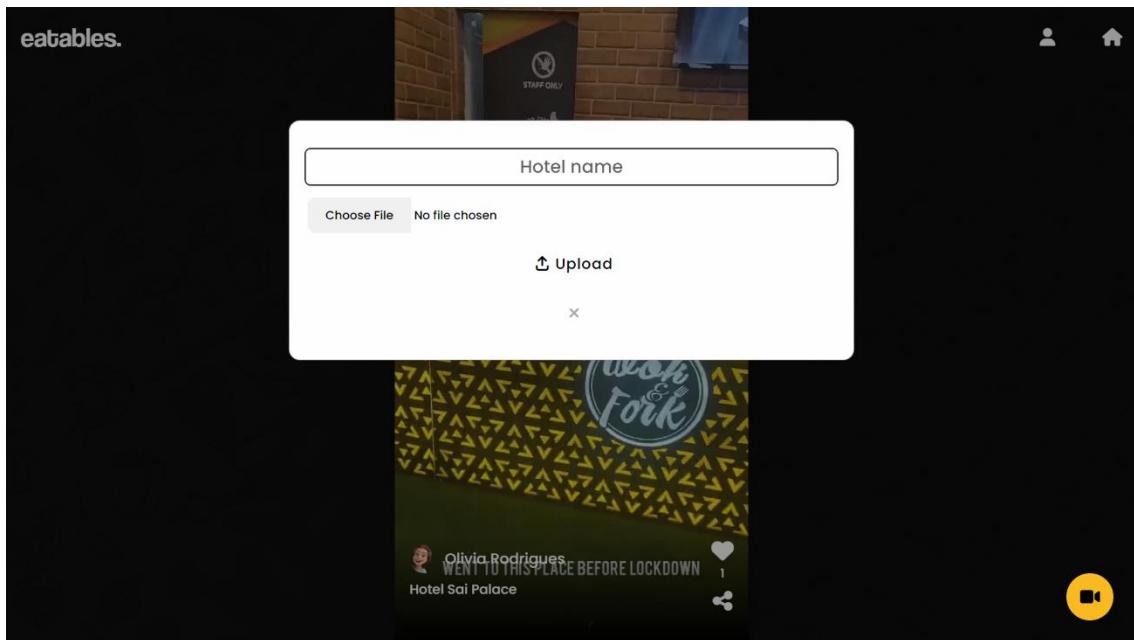
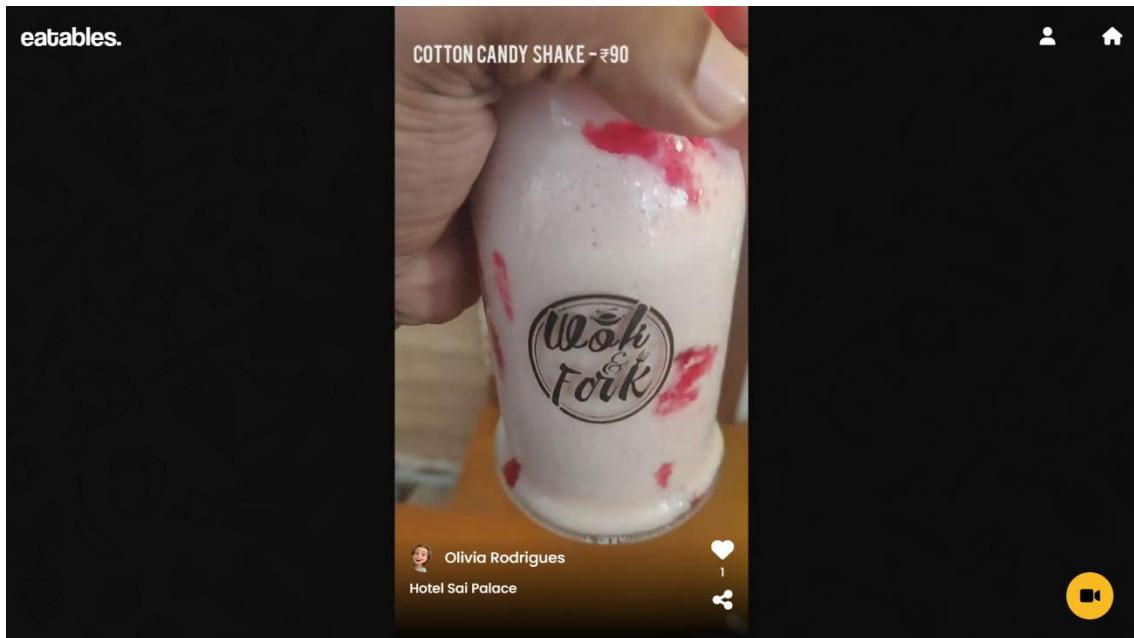
athul 13 hours ago

This pizza is very poor pizza.No quality and prize is very high.



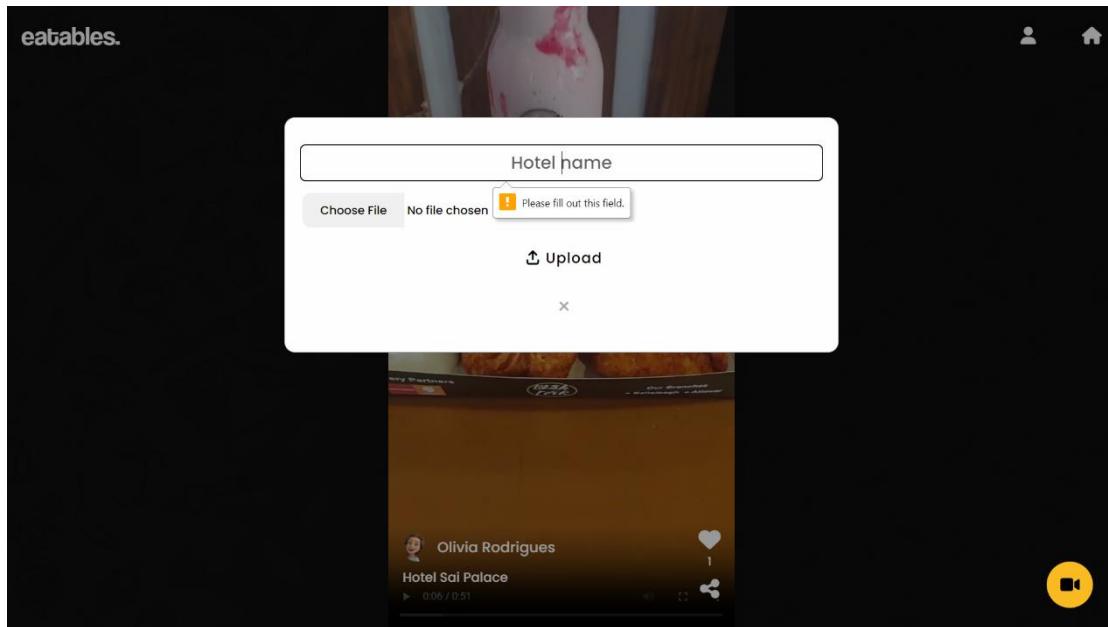
### 5.3.7.Drops page

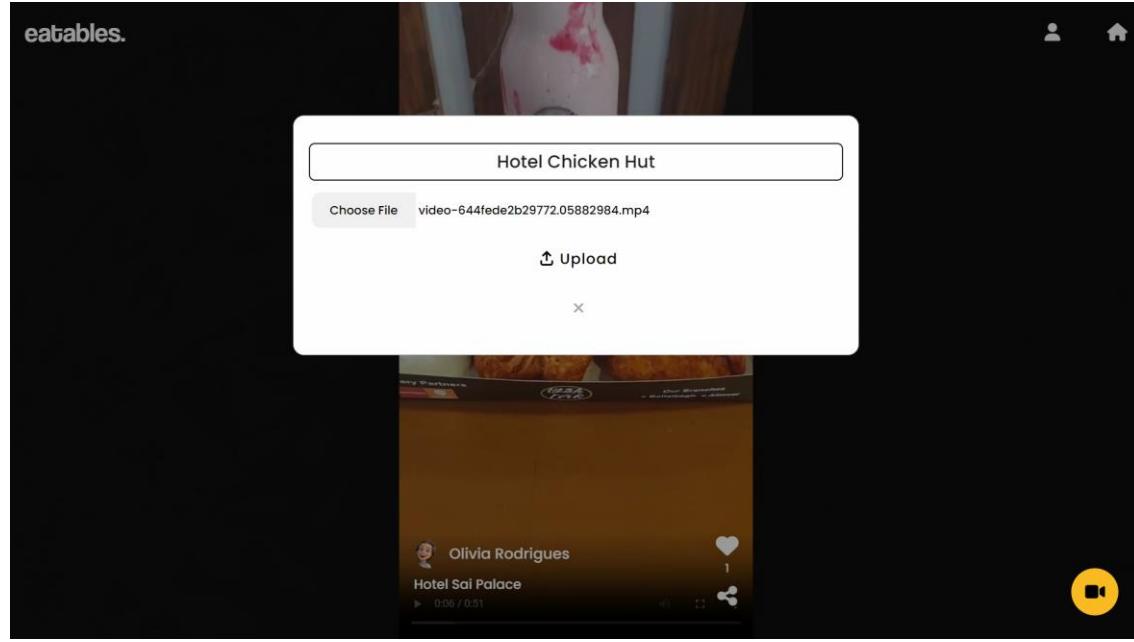
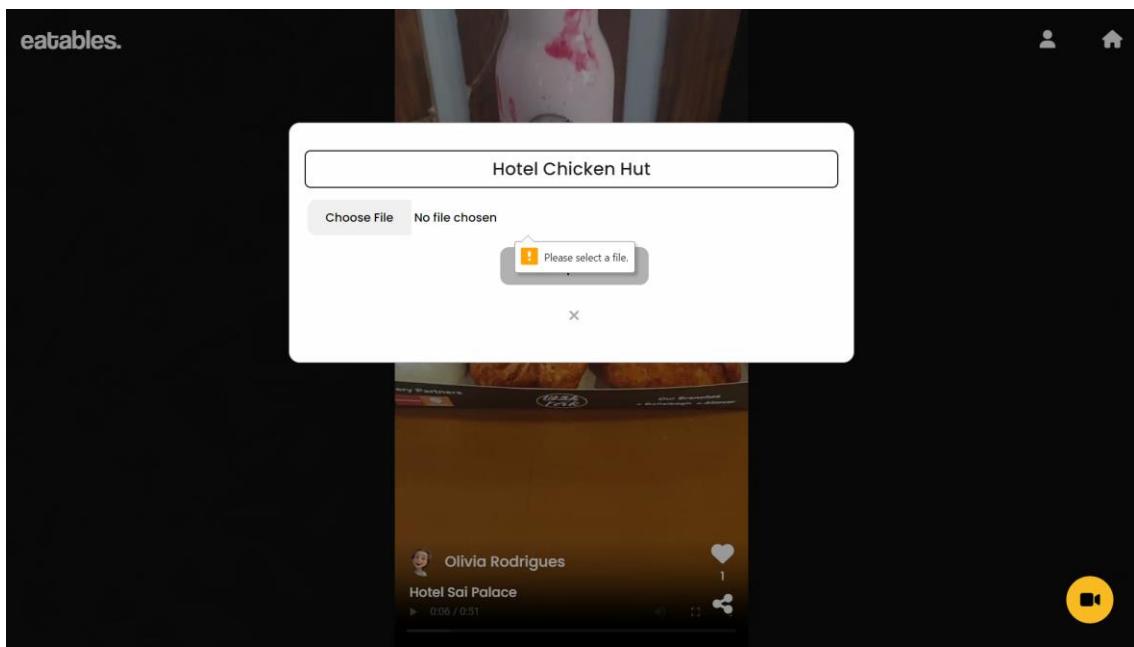
Test Case ID	Test Condition	Expected Output	Result
1	If user clicks on eatables	Redirect to home page	Success
2	If user clicks on profile icon	Redirect to user profile page	Success
3	If user clicks in home icon	Redirect to index page	Success
4	If user clicks on heart icon	Like/dislike the video	Success
5	If user click/tap in the video	Play/pause the video	Success
6	If user click on the video icon	Open upload video window	Success



### 5.3.8.Upload drop window

Test Case ID	Test Condition	Expected Output	Result
1	If tried to upload empty fields	Display please fill out this field	Success
2	If submitted without uploading file	Display please fill out this field	Success
3	If submitted without restaurant name	Display please fill out this field	Success
4	If submitted with valid field data	Upload the video and redirect to drops page	Success

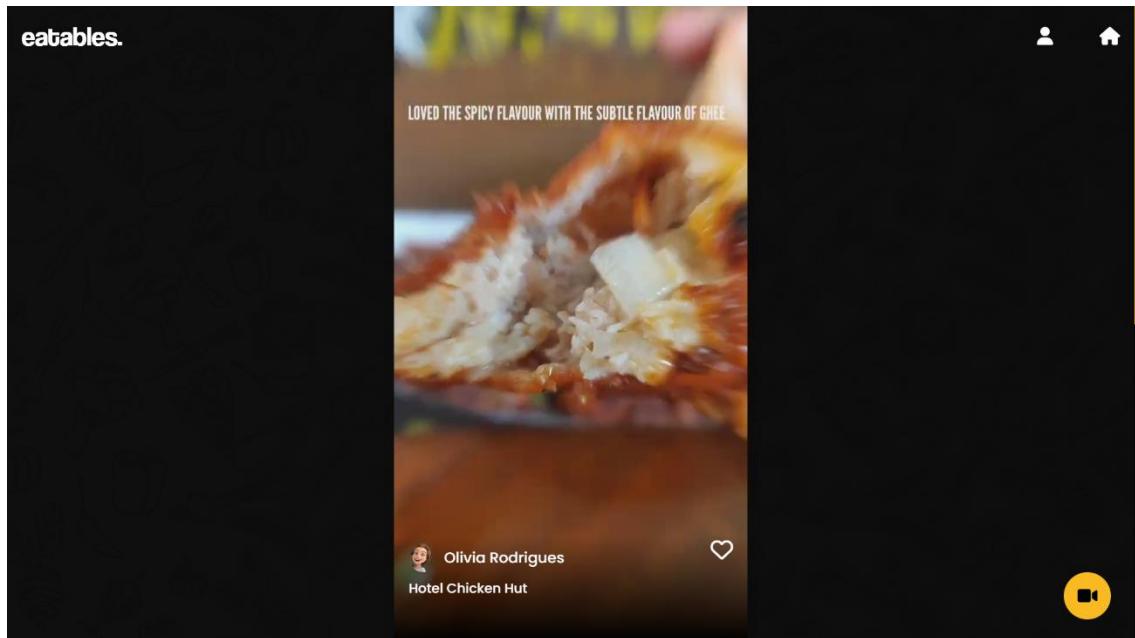




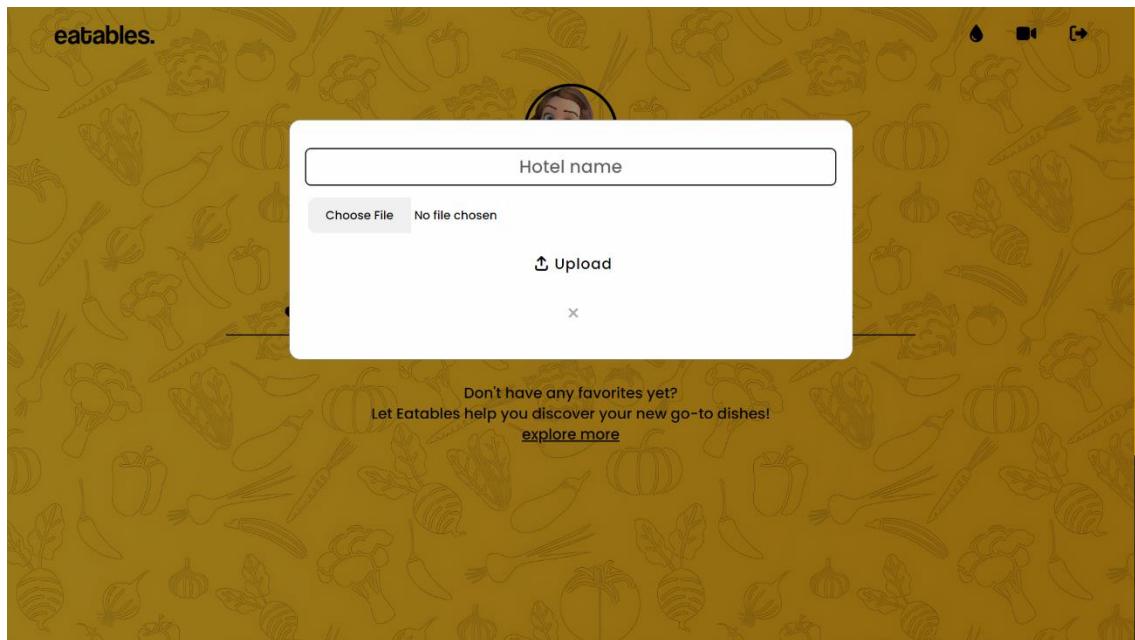
### 5.3.9.User profile

<b>Test Case ID</b>	<b>Test Condition</b>	<b>Expected Output</b>	<b>Result</b>
1	If clicked on eatables	Redirect to home page	Success
2	If clicked on drop icon	Redirect to drops page	Success
3	If clicked on video icon	Open drop upload window	Success
4	If clicked on logout icon	Log out and redirect to login page	Success
5	If clicked on camera icon	Redirect to edit profile page	Success
6	If clicked on edit profile button	Redirect to edit profile page	Success
7	If clicked on Favorites	Display items in user's favorite list	Success
8	If clicked on the delete icon on the favorite item	Remove from favorite	Success
9	If clicked on Drops	Display drops uploaded by the user	Success
10	If clicked on Reviews	Display reviews uploaded by the user	Success
11	If clicked on delete icon in reviews	Delete the review and redirect to the same page	Success
12	If clicked on submit restaurant in footer	Redirect to restaurant suggest form page	Success

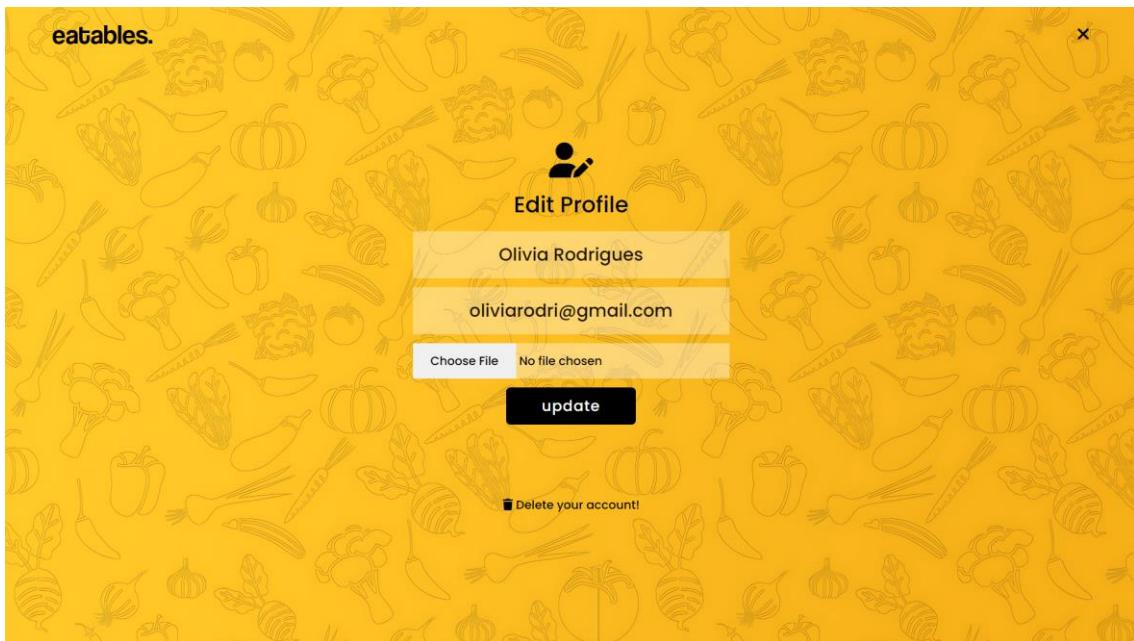
User profile>>Drops Icon



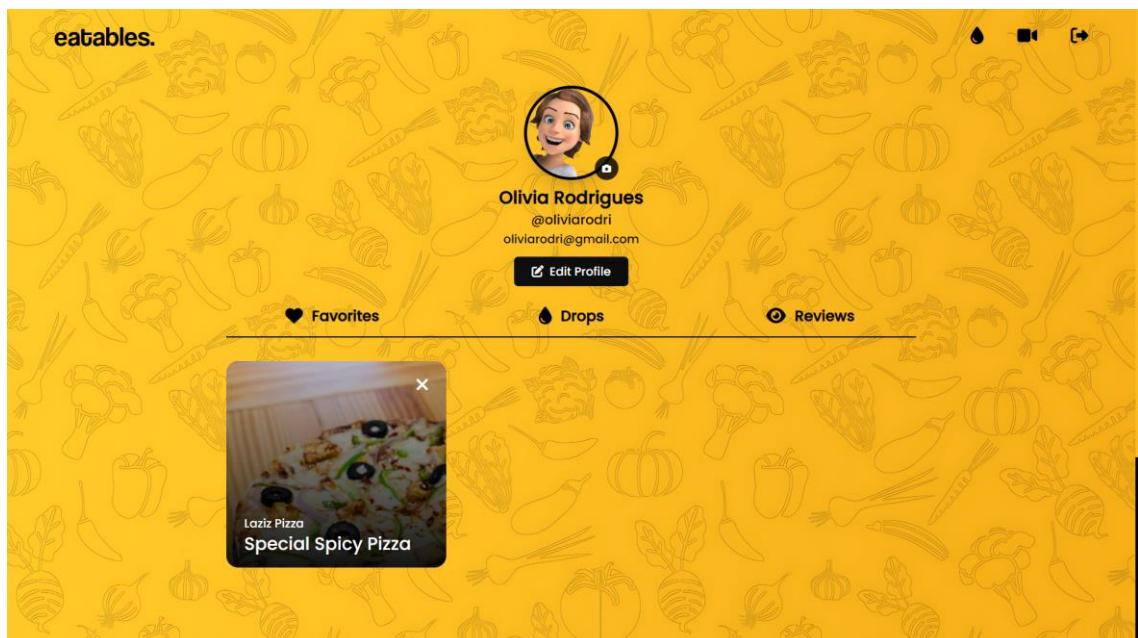
User profile>>Upload



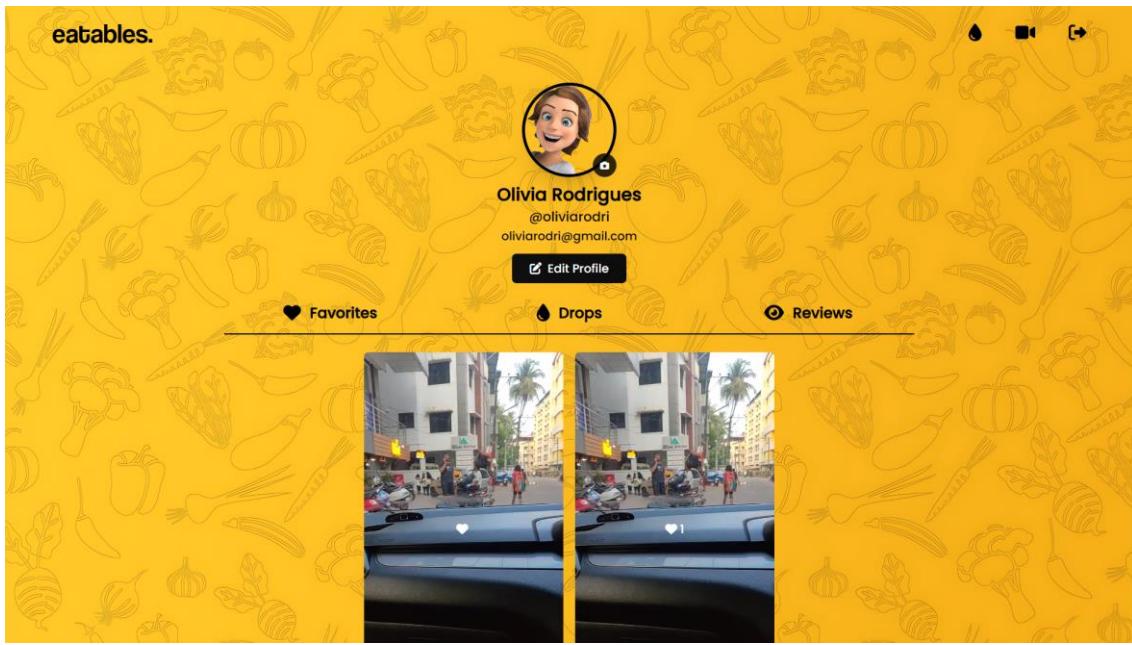
User profile>>Edit profile



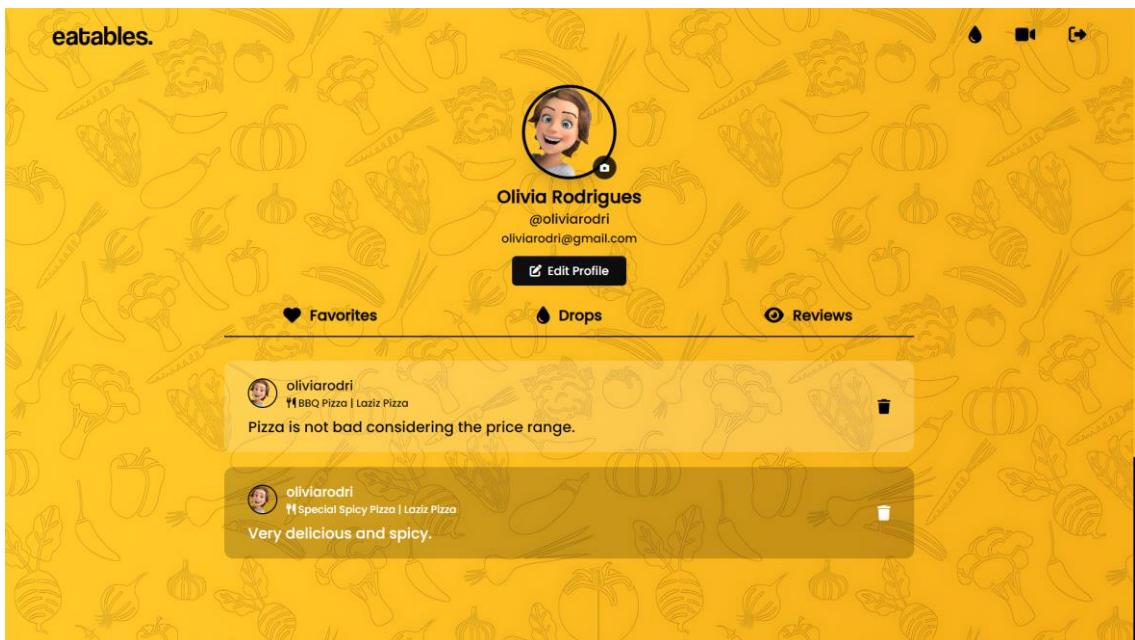
User profile>>Favorites



## User profile>>Drops

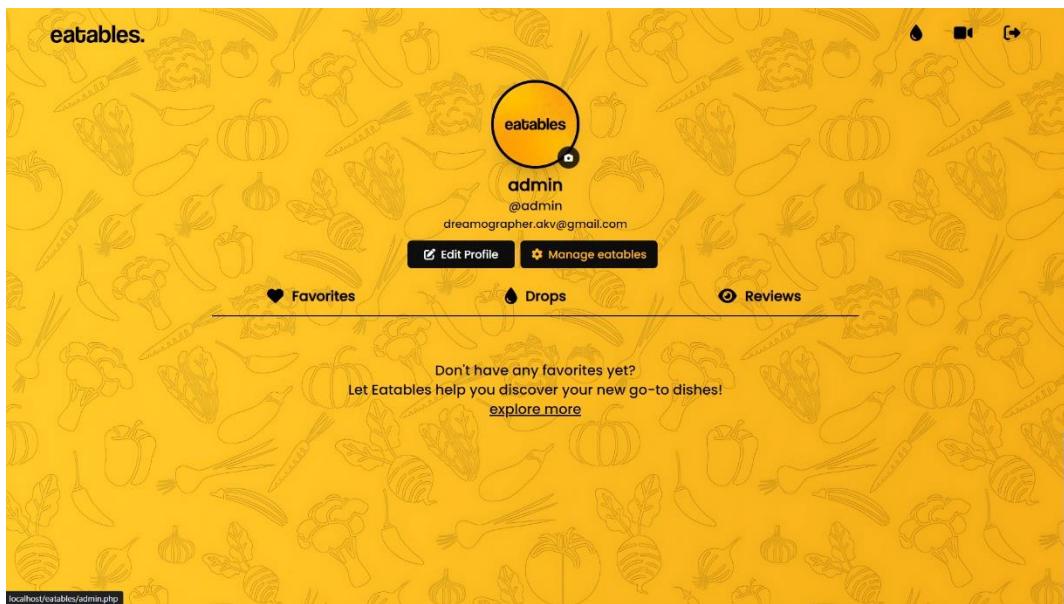


## User profile>>Reviews



### 5.3.10.Admin profile

Test Case ID	Test Condition	Expected Output	Result
1	If clicked on Manage eatables	Redirect to admin panel	Success



Admin profile>>Manage Eatables :

Hotel Name	Location	Rating	Links	Description	Action
Hotel Chicken Hut	https://www.gmap.com/ofagb24b8	4	https://www.zomato.com/ofagb24b8	Hotel in kankanady	

### 5.3.11.Admin panel

Test Case ID	Test Condition	Expected Output	Result
1	If clicked on add new restaurant	Display form to add new restaurant	Success
2	If clicked on manage restaurant	Display list of all restaurants with edit and delete button	Success
3	If clicked on add item	Display form to add new item	Success
4	If clicked on user suggestion	Display all restaurant suggestions submitted by different users	Success
5	If clicked on admin logo	Redirect to admin profile page	Success

Admin Panel>>ADD NEW RESTARANT :

The screenshot shows the Admin Panel interface for adding a new restaurant. At the top, there are four yellow buttons: '+ ADD NEW RESTAURANT', 'MANAGE RESTAURANT', '+ ADD ITEM', and '+ User suggestions'. Below these buttons is a navigation bar with icons for Home, Admin, Logout, and Help. The main content area is titled 'ADD NEW RESTAURANT'. It contains the following fields:

- Hotel name: [Input field]
- Location: [Dropdown menu set to 'Manglore']
- Location (Gmap location link): [Input field]
- Rating(1-5): [Input field]
- link (zomato link): [Input field]
- description: [Input field]

At the bottom of the form is a black 'Add' button.

## Admin Panel>>MANAGE RESTAURANT:

The screenshot shows the 'Manage Restaurant' section of the admin panel. At the top, there are four buttons: '+ ADD NEW RESTAURANT' (yellow), 'MANAGE RESTAURANT' (orange), '+ ADD ITEM' (yellow), and '+ User suggestions' (yellow). Below these buttons is a table with columns: 'Hotel Name', 'Edit', and 'Delete'. The table lists eight restaurants: Laziz Pizza, Hamburg Street Food Cafe, Bamboo Restaurant, CKK Kitchens Of Maharaja, Hotel Sai Palace, Grameen Kulfi, Momos Hut, and Keventers - Milkshakes & Desse. Each row has an 'Edit' button (with a pencil icon) and a 'Delete' button (with a trash bin icon). The URL 'localhost/estables/admin/manage\_restaurant.php' is visible at the bottom of the page.

Hotel Name	Edit	Delete
Laziz Pizza		
Hamburg Street Food Cafe		
Bamboo Restaurant		
CKK Kitchens Of Maharaja		
Hotel Sai Palace		
Grameen Kulfi		
Momos Hut		
Keventers - Milkshakes & Desse		
Danish Bamboo House		

## Admin Panel>>ADD ITEM:

The screenshot shows the 'Add Item' form for the 'Laziz Pizza' hotel. At the top, there are four buttons: '+ ADD NEW RESTAURANT' (yellow), 'MANAGE RESTAURANT' (orange), '+ ADD ITEM' (yellow), and '+ User suggestions' (yellow). Below these buttons is a dropdown menu labeled 'HOTEL:' with 'Laziz Pizza' selected. There are two input fields: 'item name:' and 'item price:', both currently empty. At the bottom of the form is a large black 'Add' button.

Admin Panel>>USER SUGGESTIONS :

Hotel Name	Location	Rating	Links	Description	Action
Hotel Chicken Hut	<a href="https://www.gmap.com/ofagb24b8">https://www.gmap.com/ofagb24b8</a>	4	<a href="https://www.zomato.com/ofagb24b8">https://www.zomato.com/ofagb24b8</a>	Hotel in kankanady	

### 5.3.12.Add new restaurant form

Test Case ID	Test Condition	Expected Output	Result
1	If hotel name field empty	Display please fill out this field	Success
2	If location field empty	Display please fill out this field	Success
3	If rating field empty	Display please fill out this field	Success
4	If link field empty	Display please fill out this field	Success
5	If description field empty	Display please fill out this field	Success
6	If all fields are filled and submitted	Display restaurant added message and redirect to admin panel	Success

+ ADD NEW RESTAURANT    MANAGE RESTAURANT    + ADD ITEM    + User suggestions

Hotel name :

Loco ! Please fill out this field.  
Manglore

Location (Gmap location link):  
<https://www.gmap.com/>

Rating(1-5):  
4

link (zomato link):  
<https://www.zomato.com>

description:  
Veg and non veg

Add

+ ADD NEW RESTAURANT    MANAGE RESTAURANT    + ADD ITEM    + User suggestions

Hotel name :  Hotel Chicken Hut

Location:  
Manglore

Location (Gmap location link):

Rating(1-5): ! Please fill out this field.  
4

link (zomato link):  
<https://www.zomato.com>

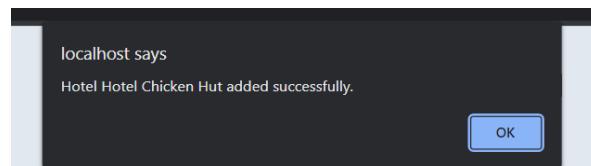
description:  
Veg and non veg

Add

The screenshot shows a user interface for adding a new restaurant. At the top, there are four buttons: '+ ADD NEW RESTAURANT' (orange), 'MANAGE RESTAURANT' (blue), '+ ADD ITEM' (orange), and '+ User suggestions' (orange). Below these, the form fields are displayed:

- Hotel name:** Hotel Chicken Hut
- Location:** Manglore
- Location (Gmap location link):** <https://www.gmap.com/>
- Rating(1-5):** 4
- link (zomato link):** [Empty input field]
- description:** Veg and non veg (This field has a red border indicating it is required)

A black 'Add' button is located at the bottom right of the form.



### 5.3.13. Manage restaurant form

Test Case ID	Test Condition	Expected Output	Result
1	If clicked on edit icon	Redirect to edit restaurant form page	Success
2	If field empty	Display please fill out this field	success
3	If fields are valid	Display update successful	Success
4	If clicked on delete icon	Delete the restaurant and redirect to same page	Success

**admin**

+ ADD NEW RESTAURANT    MANAGE RESTAURANT    + ADD ITEM    + User suggestions

Hotel Name	Edit	Delete
Laziz Pizza	<input checked="" type="checkbox"/>	
Hamburg Street Food Cafe	<input checked="" type="checkbox"/>	
Bamboo Restaurant	<input checked="" type="checkbox"/>	
CKK Kitchens Of Maharaja	<input checked="" type="checkbox"/>	
Hotel Sai Palace	<input checked="" type="checkbox"/>	
Grameen Kulfi	<input checked="" type="checkbox"/>	
Momos Hut	<input checked="" type="checkbox"/>	
Keventers - Milkshakes & Desse	<input checked="" type="checkbox"/>	
Danish Bamboo House	<input checked="" type="checkbox"/>	

localhost/estables/admin/edit.php?name=Laziz Pizza

Edit restaurant details.

Hotel name: Laziz

Location: Manglore

Location (lat and long):  Please fill out this field.

Rating(1-5):

link (zomato link): <https://www.zomato.com>

description:

Add

Edit restaurant details.

Hotel name : Laziz

Location: Mangalore

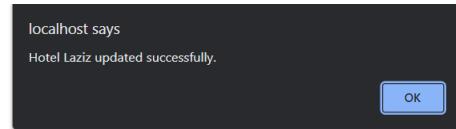
Location (lat and long): 3902,1233

Rating(1-5): 4

link (zomato link): <https://www.zomato.com>

description: Hotel Laziz Pizza

Add



### 5.3.14.Add item form

Test Case ID	Test Condition	Expected Output	Result
1	If field empty	Display please fill out this field	Success
2	If all fields are valid	Display item added message	Success

 admin

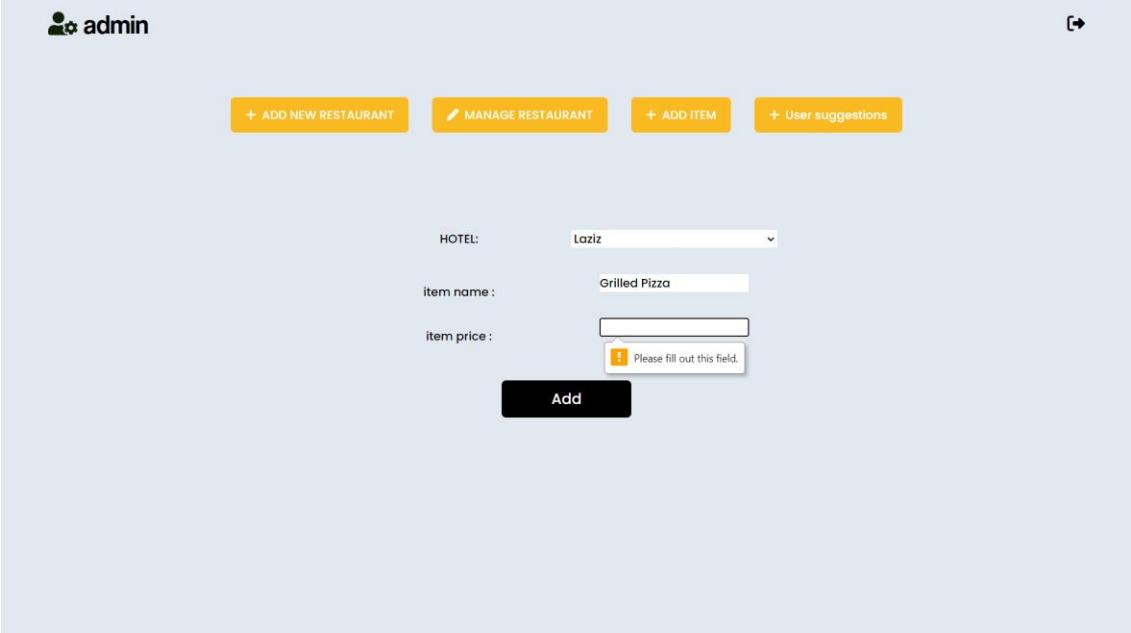
[+ ADD NEW RESTAURANT](#) [MANAGE RESTAURANT](#) [+ ADD ITEM](#) [+ User suggestions](#)

HOTEL: Laziz

item name: Grilled Pizza

item price:  Please fill out this field.

**Add**



 admin

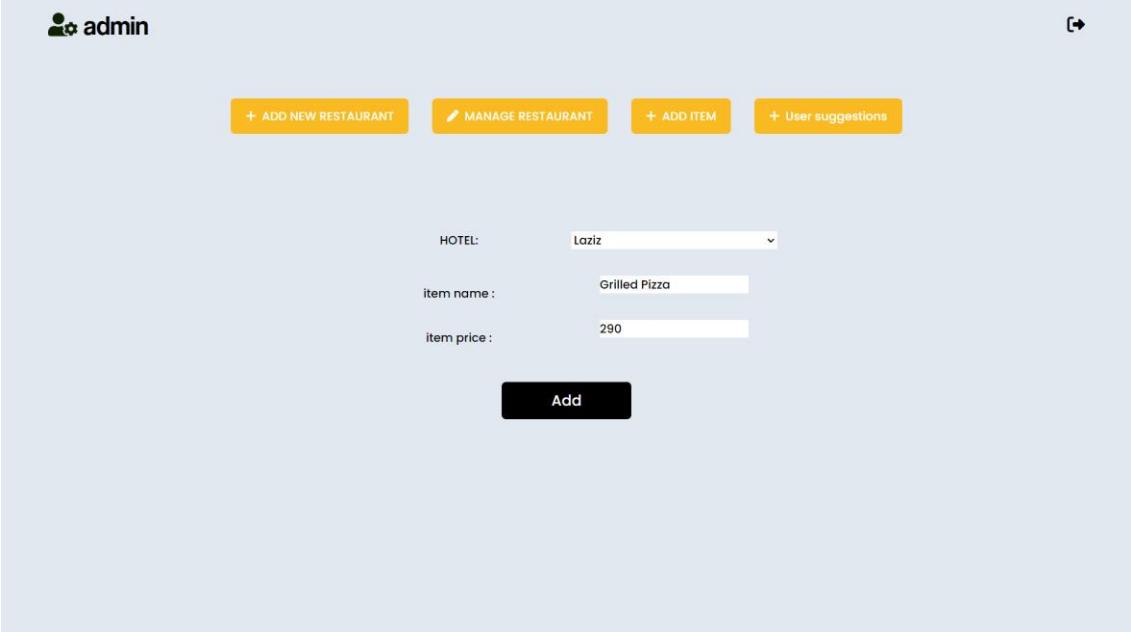
[+ ADD NEW RESTAURANT](#) [MANAGE RESTAURANT](#) [+ ADD ITEM](#) [+ User suggestions](#)

HOTEL: Laziz

item name: Grilled Pizza

item price: 290

**Add**

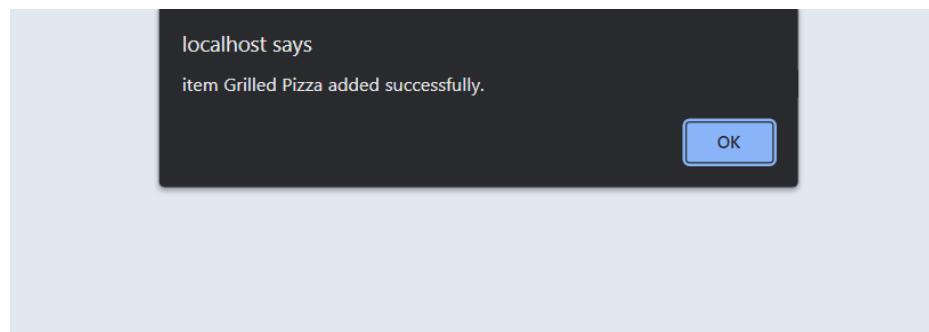


### 5.3.15.User suggestion review

Test Case ID	Test Condition	Expected Output	Result
1	If clicked on green icon	Redirect to add hotel form page along with hotel details	Success
2	If clicked on red icon	Reject/Delete the suggestion	Success

The screenshot shows a web application interface for managing restaurants. At the top, there is a navigation bar with a user icon labeled "admin" and a back arrow icon. Below the navigation bar are four yellow buttons with black text: "+ ADD NEW RESTAURANT", "MANAGE RESTAURANT", "+ ADD ITEM", and "+ User suggestions". The main content area has a light gray background. At the top of this area, the text "New hotel suggestions." is displayed. Below this, there is a table with a header row containing columns for "Hotel Name", "Location", "Rating", "Links", "Description", and "Action". A single data row is shown in the table:

Hotel Name	Location	Rating	Links	Description	Action
Hot Sulaimani	<a href="https://www.gmap.com/ofwcf837wg">https://www.gmap.com/ofwcf837wg</a>	4	<a href="https://www.zomato.com/ofwcf837wg">https://www.zomato.com/ofwcf837wg</a>	Veg and Non veg hotel	



 admin

**Hotel name :** Hot Sulaimani

**Location:** Manglore

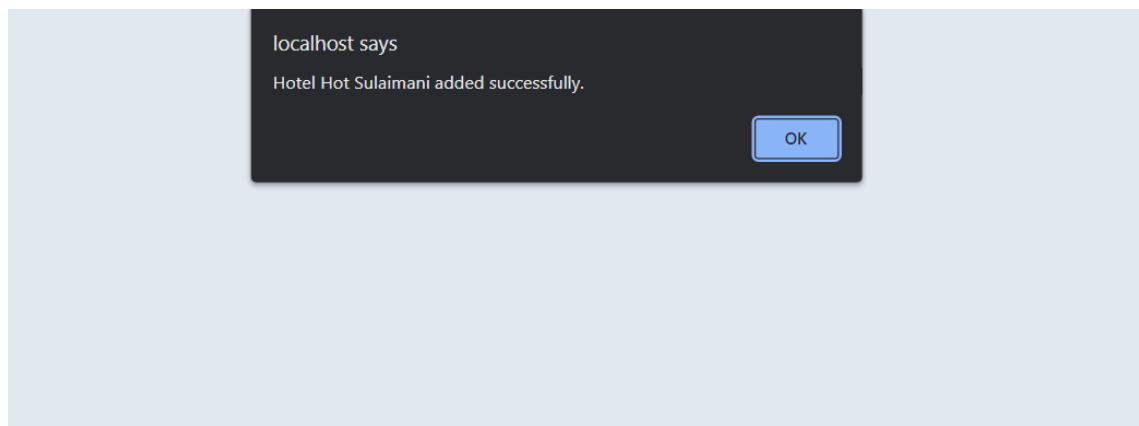
**Location (Gmap location link):** <https://www.gmap.com/>

**Rating(1-5):** 4

**link (zomato link):** <https://www.zomato.com>

**description:** Veg and Non veg hotel

**Add**



## CONCLUSION

In conclusion, this project aimed to create a food review website called "Eatables" that provides users with a platform to explore and review various food items. The project successfully achieved its objectives by implementing key features such as user registration and authentication, food item listing, user reviews, and filtering based on price.

Throughout the development process, careful attention was given to creating an intuitive and user-friendly interface. The website's design and layout were optimized to enhance the user experience and facilitate easy navigation. Additionally, security measures, such as user authentication and data validation, were implemented to ensure the privacy and integrity of user information.

The integration of a database management system allowed for efficient storage and retrieval of food item data, enabling seamless filtering based on price. Users can now easily browse and search for food items within their desired price range, enhancing their overall experience on the website.

The project also implemented interactive features, such as the ability for users to write reviews for food items. This encourages user engagement and provides valuable insights for other visitors to the website.

Overall, the "Eatables" project showcases the successful implementation of a food review website, demonstrating the potential for providing a valuable platform for food enthusiasts to discover and review various food items.