

**Indian Institute of Information Technology,  
Guwahati**

**B.Tech Computer Science**

**Software Engineering Project**

**FoodHub - Online Food Ordering System**

**Testing Report**

**Group 15:**

Swaraj Parth (1901205)  
Soppa Prasunna (1901199)  
Priyanshi (1901149)  
Shyam Marjit (1901195)

Link to the project :- <https://github.com/swarajparth/FoodHub>

# Manual Testing

The type of testing used in this project is **Black box Testing**.

It is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications.



## Functional Requirements:-

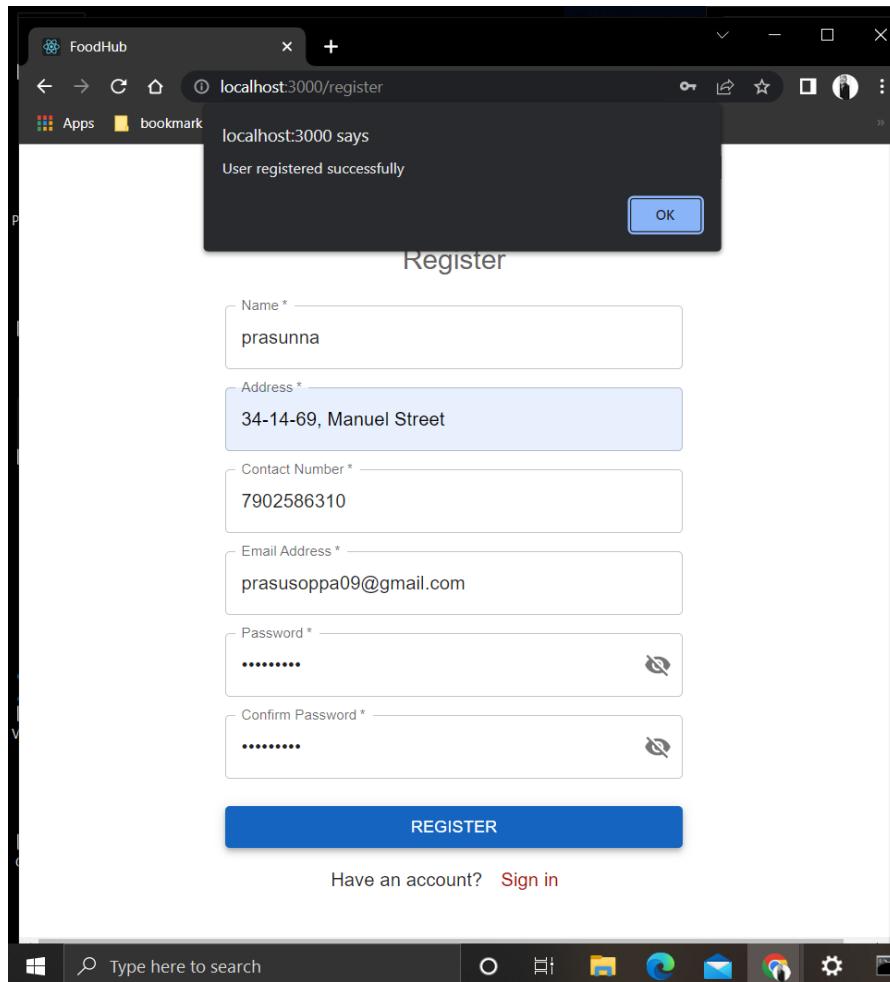
1. Users must have a valid Username, address, email and password to register thus creating their individual profiles:

### Case 1: Correct Input

Input: Correct values are given

Expected Output: User registered successfully.

Output:



## Before registering:

The screenshot shows the MongoDB Cloud interface for a project named "Swaraj's Org - 2022...". The left sidebar has sections for Deployment (Database: Data Lake), Data Services (Triggers, Data API PREVIEW), and Security (Database Access, Network Access, Advanced). The main area shows a database named "foodhub" with a "users" collection. The "Find" tab is selected, displaying two documents:

```
_id: ObjectId("6247fd35d2498fdc09af84b8")
name: "Swaraj"
address: "Bongona"
mobile: 981234560
email: "swara@gmail.com"
password: "$2a$12$Ej14gP6ukH/tj4J.h6Qt4O2l2pF0DtwRv1fgCjHn0Y7Zyan4zG"
confirm_password: "$2a$12$Ej14gP6ukH/tj4J.h6Qt4O2l2pF0DtwRv1fgCjHn0Y7Zyan4zG"
__v: 0

_id: ObjectId("624eb6864f7ad5008febe96")
name: "Swaraj Parth"
```

System Status: All Good

## After registering:

The screenshot shows the same MongoDB Cloud interface after a new user has been registered. The "users" collection now contains three documents:

```
_id: ObjectId("624f55d096bd5f548bae30")
name: "prasanna"
address: "34-14-69, Manvel Street"
mobile: 7902586310
email: "prasannapoppo0@gmail.com"
password: "$2a$12$5j3ckhoxSdfuH5b0LhgB0q4xALHdtck1FwzR.KLVCQpq13e/J1Xn"
confirm_password: "$2a$12$5j3ckhoxSdfuH5b0LhgB0q4xALHdtck1FwzR.KLVCQpq13e/J1Xn"
__v: 0

_id: ObjectId("6247fd35d2498fdc09af84b8")
name: "Swaraj"
address: "Bongona"
mobile: 981234560
email: "swara@gmail.com"
password: "$2a$12$Ej14gP6ukH/tj4J.h6Qt4O2l2pF0DtwRv1fgCjHn0Y7Zyan4zG"
confirm_password: "$2a$12$Ej14gP6ukH/tj4J.h6Qt4O2l2pF0DtwRv1fgCjHn0Y7Zyan4zG"
__v: 0

_id: ObjectId("624eb6864f7ad5008febe96")
name: "Swaraj Parth"
__v: 0
```

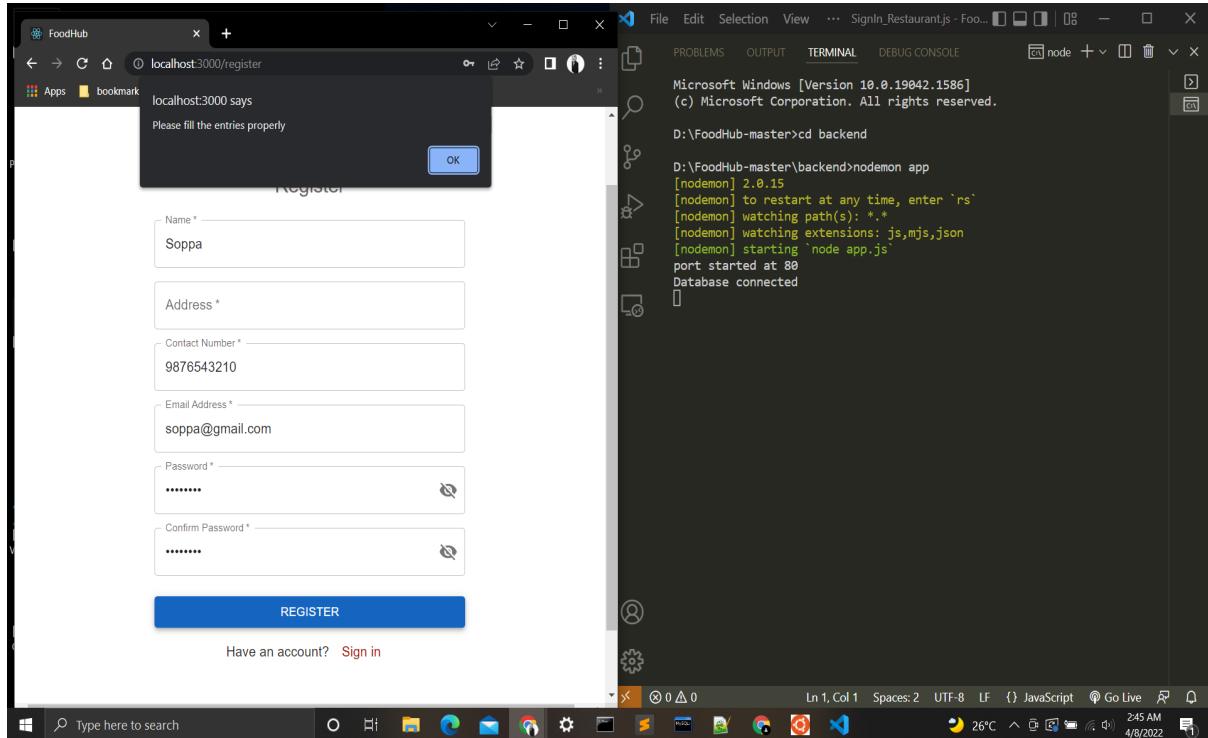
System Status: All Good

## Case 2: Incorrect Input

Input: One of the i/p values is missing.

Expected Output: Fill the entries properly.

Output:

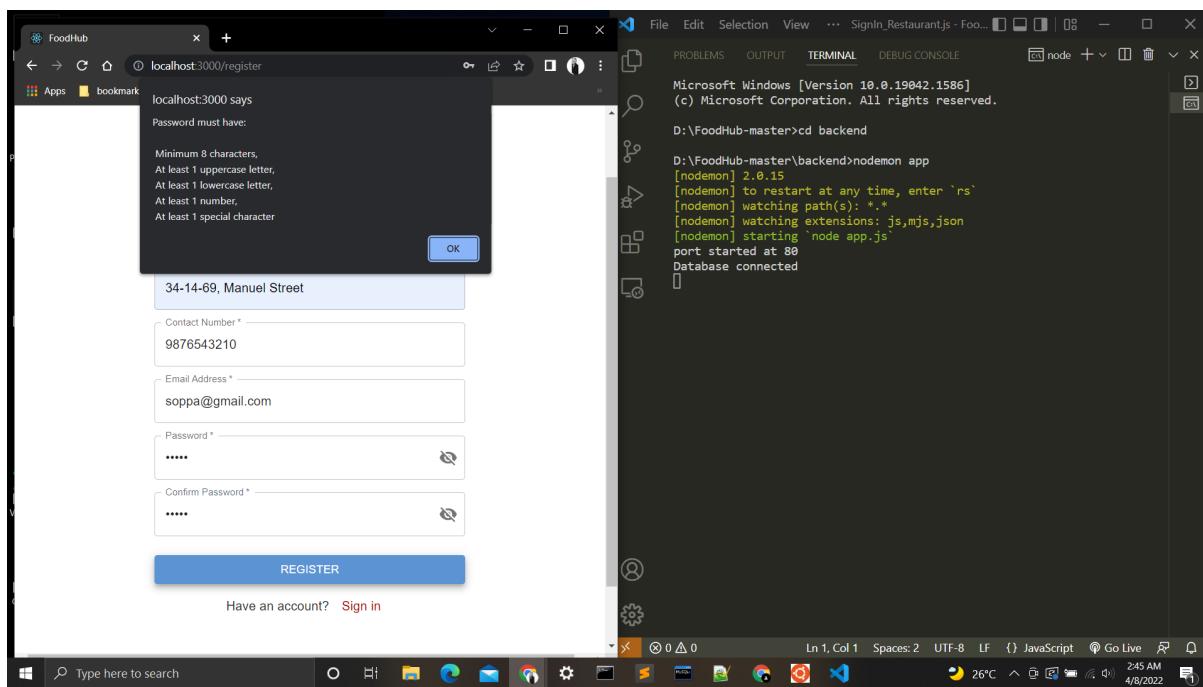


## Case 3: Incorrect password

Input: Shorter password

Expected Output: Password must have min. of 8 characters.

Output:

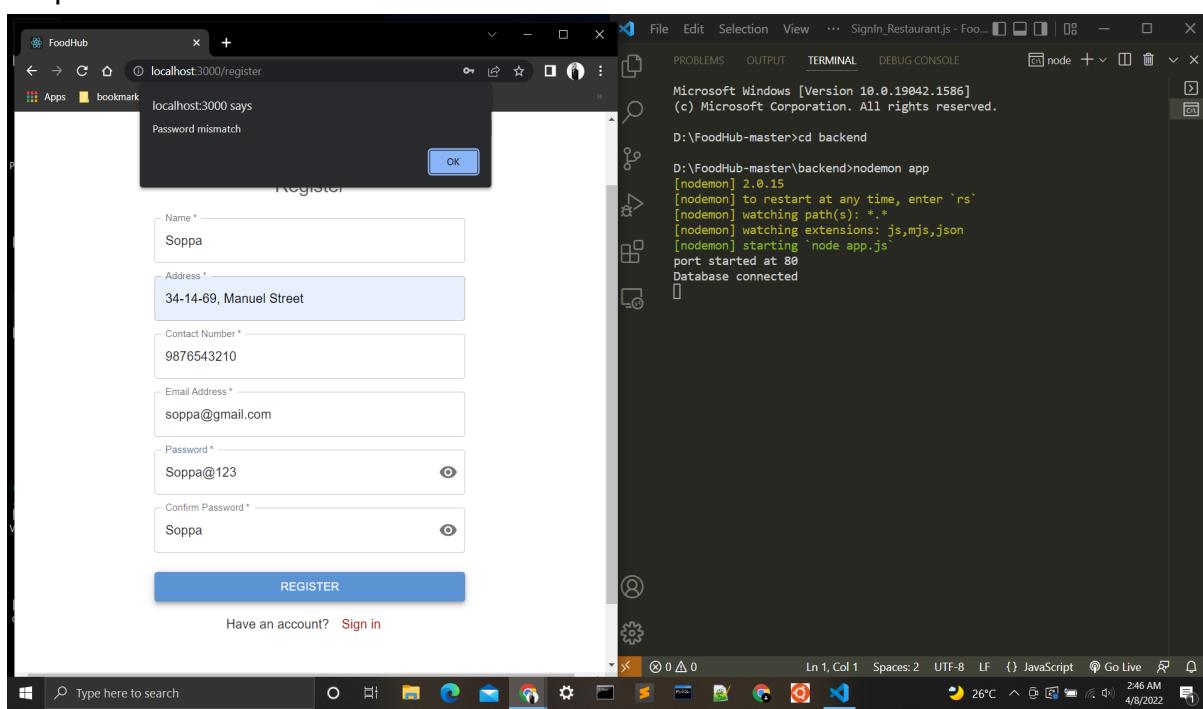


## Case 4: Password mismatch

Input: Different password is given in “confirm password” input.

Expected Output: Password mismatch

Output:



## Case 5: Invalid email address

Input: Invalid email is given.

Expected Output: Enter a valid email address.

Output:

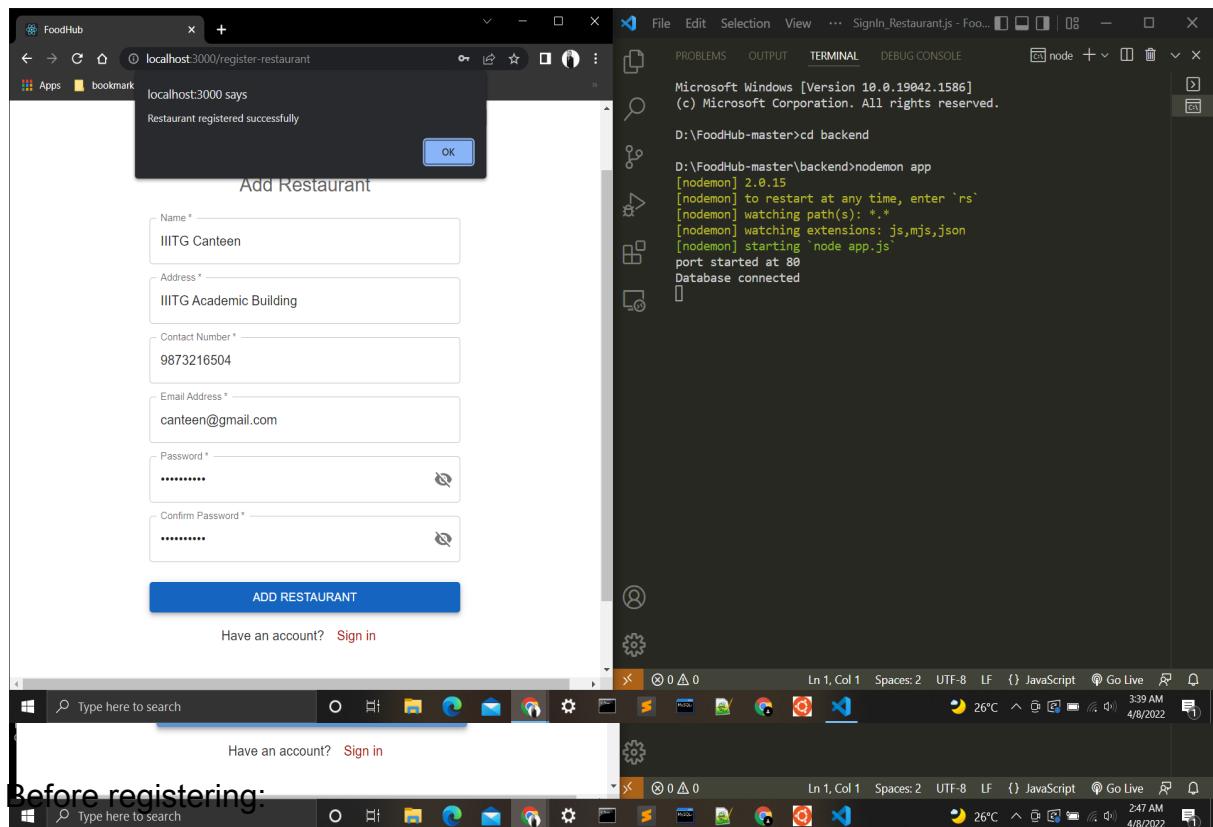
2. Restaurants must have a valid Name, address, email and password to register thus creating their individual profiles:

### Case 1: Correct Input

Input: Correct values are given

Expected Output: Restaurant registered successfully.

Output:



**foodhub.restaurants**

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

**Find** Indexes Schema Anti-Patterns Aggregation Search Indexes

**FILTER** { field: 'value' } **OPTIONS** **Apply** Reset

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("6247f5bfcd2400fd0ef04a3")
name: "Nescafe"
address: "Bongora, IITTG"
mobile: 989878130
email: "nescafe@gmail.com"
password: "$2a$12$7Qgk0PIny1Fy9Yw.1l0HabuVawKla.VpYC8a1Teex$46dbQzDmI"
confirm_password: "$2a$12$7Qgk0PIny1Fy9Yw.1l0HabuVawKla.VpYC8a1Teex$46dbQzDmI"
__v: 0
```

After registering:

**foodhub.restaurants**

STORAGE SIZE: 36KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

**Find** Indexes Schema Anti-Patterns Aggregation Search Indexes

**FILTER** { field: 'value' } **OPTIONS** **Apply** Reset

**EDIT** **CREATE** **DELETE**

```
_id: ObjectId("624f60f96bd5f85488a6e44")
name: "IITTG Canteen"
address: "IITTG Academic Building"
mobile: 9873216504
email: "canteen@gmail.com"
password: "$2a$12$Qgk0PIny1Fy9Yw.04KXeAp016VfnQ7On6q5X%IpE14plmz/oZq"
confirm_password: "$2a$12$Qgk0PIny1Fy9Yw.04KXeAp016VfnQ7On6q5X%IpE14plmz/oZq"
__v: 0
```

System Status: All Good

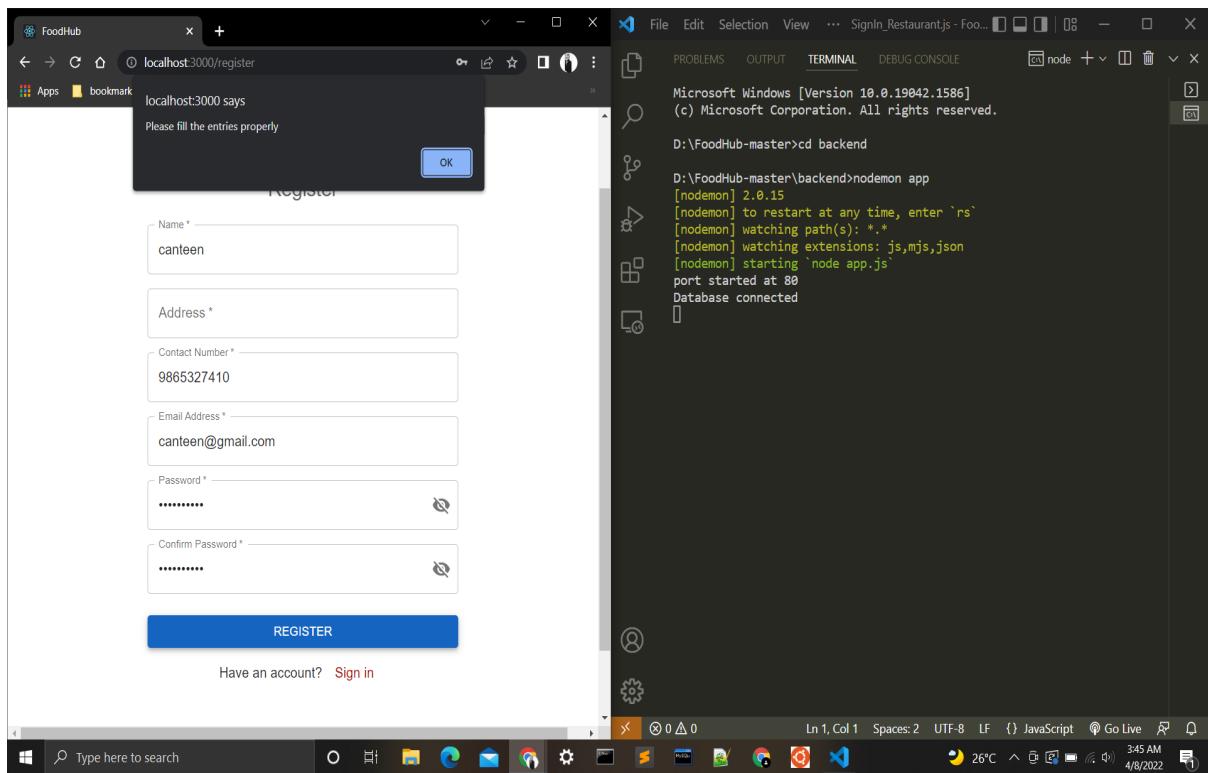
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

## Case 2: Incorrect Input

Input: One of the i/p values is missing.

Expected Output: Fill the entries properly.

Output:

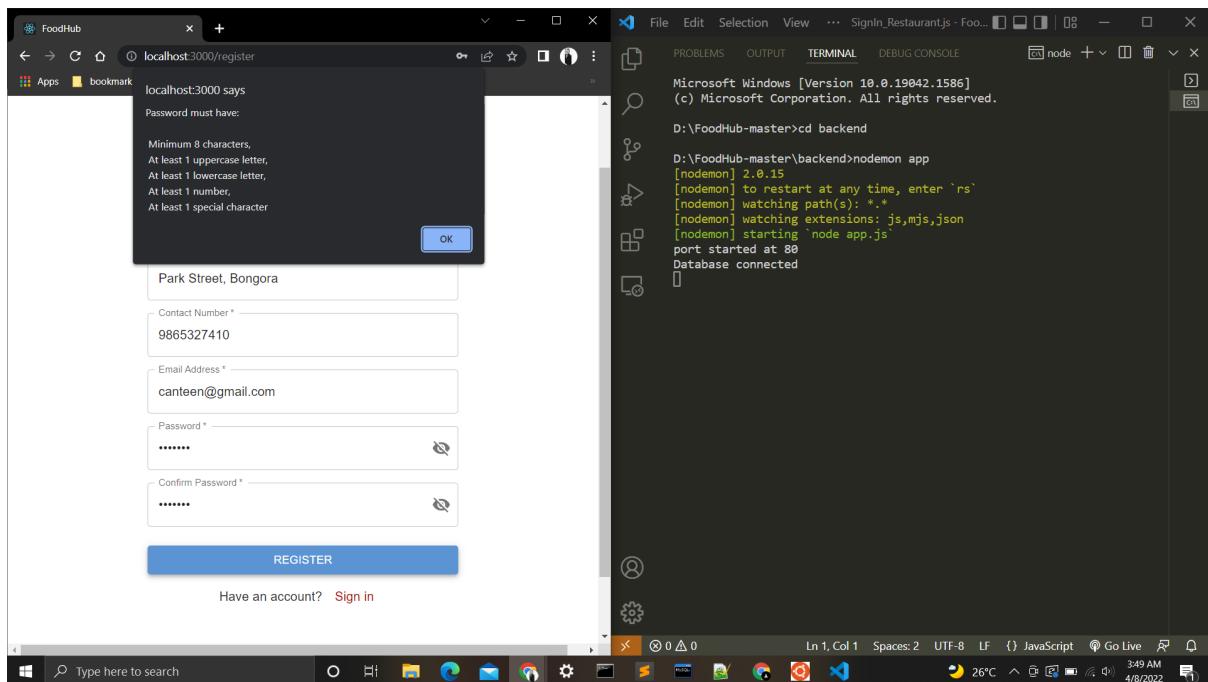


### Case 3: Incorrect password

Input: Shorter password

Expected Output: Password must have min. of 8 characters.

Output:

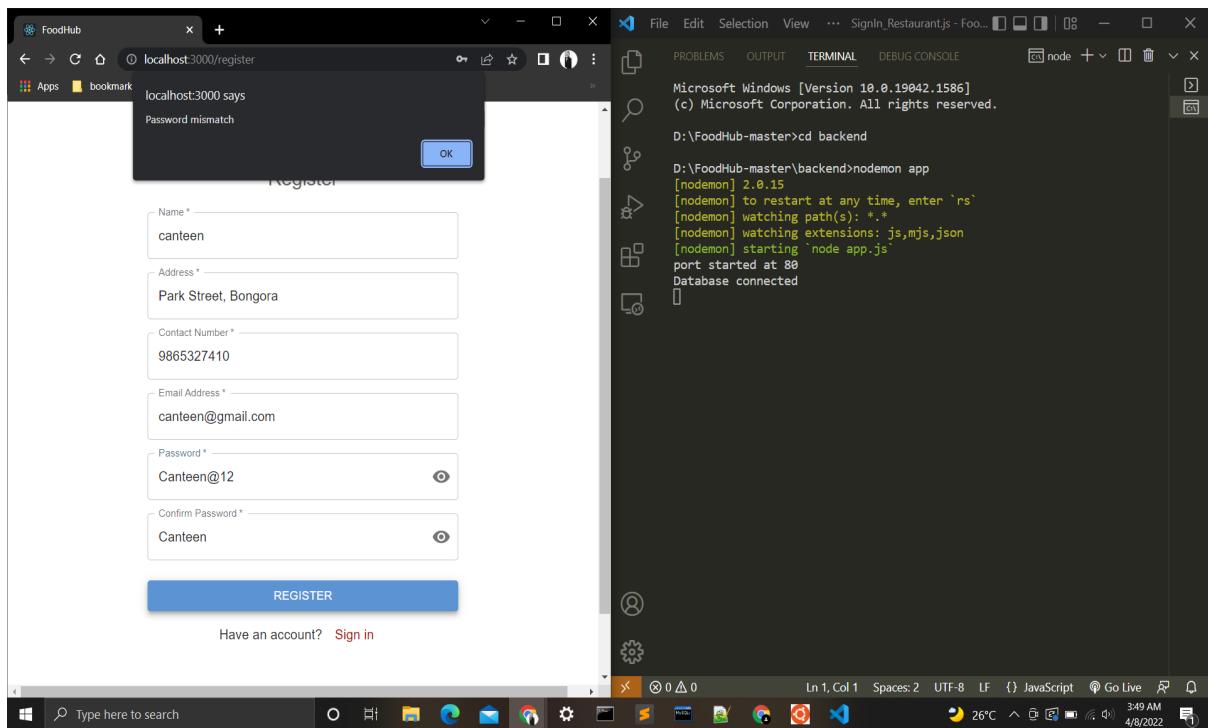


### Case 4: Password mismatch

Input: Different password is given in "confirm password" input.

Expected Output: Password mismatch

Output:

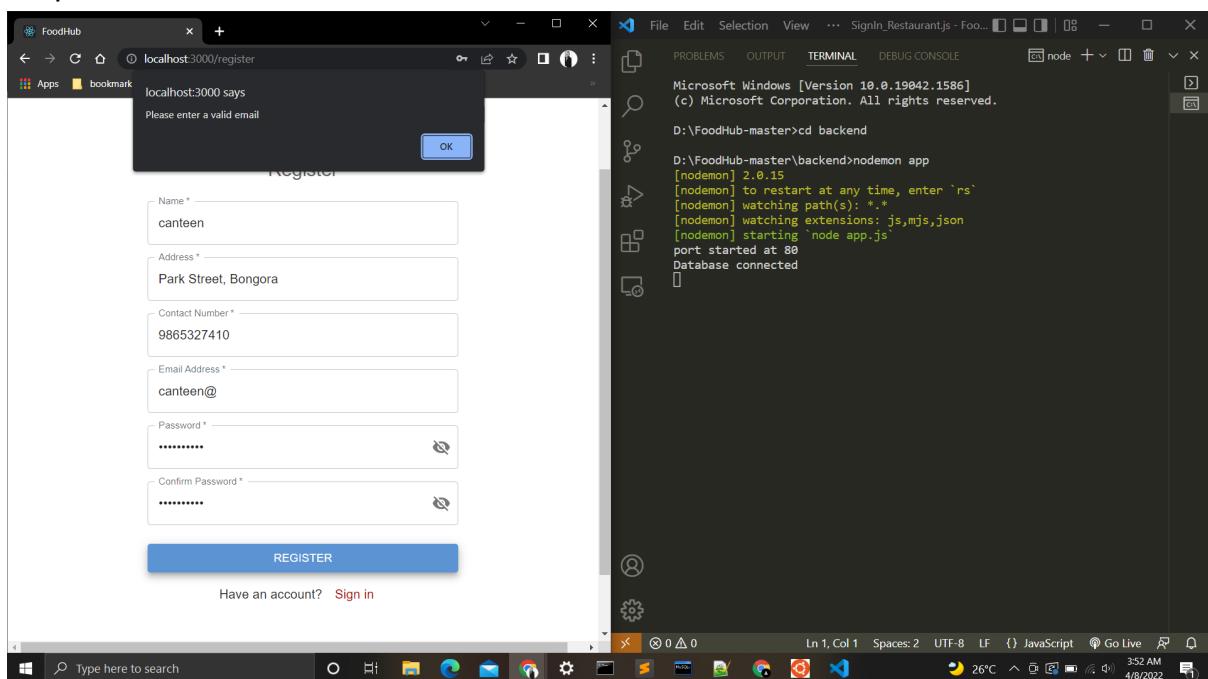


## Case 5: Invalid email address

Input: Invalid email is given.

Expected Output: Enter a valid email address.

Output:



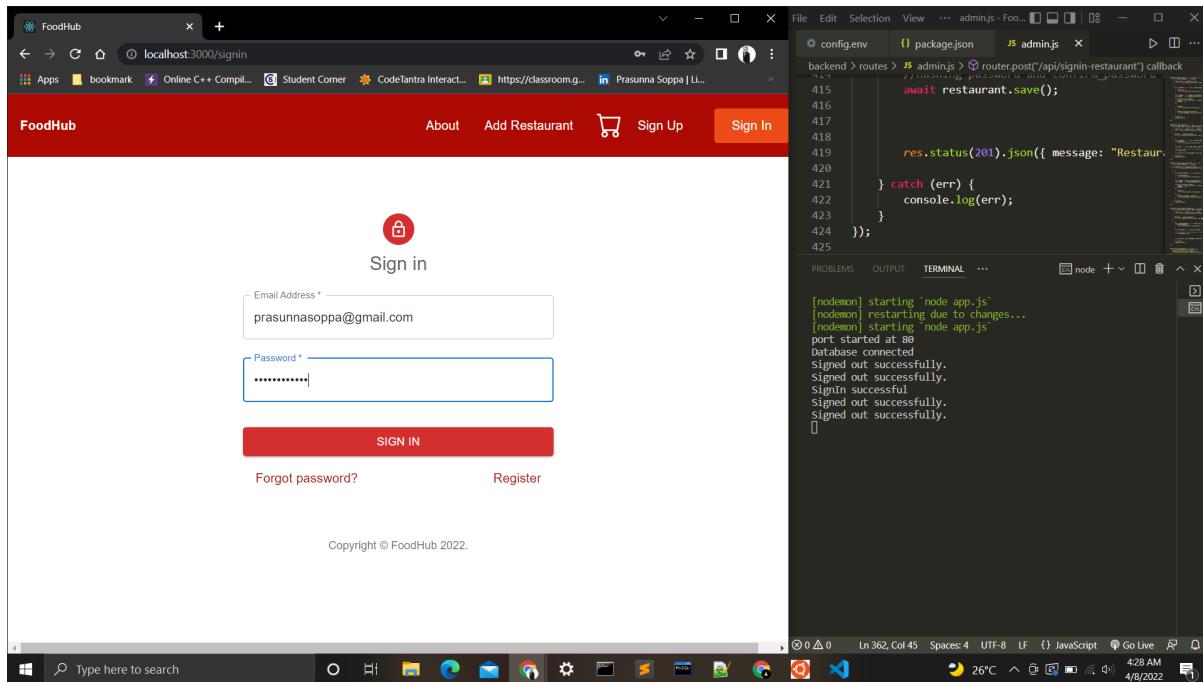
### 3. Users should enter correct credentials to login to their account:

#### Case 1: Correct input

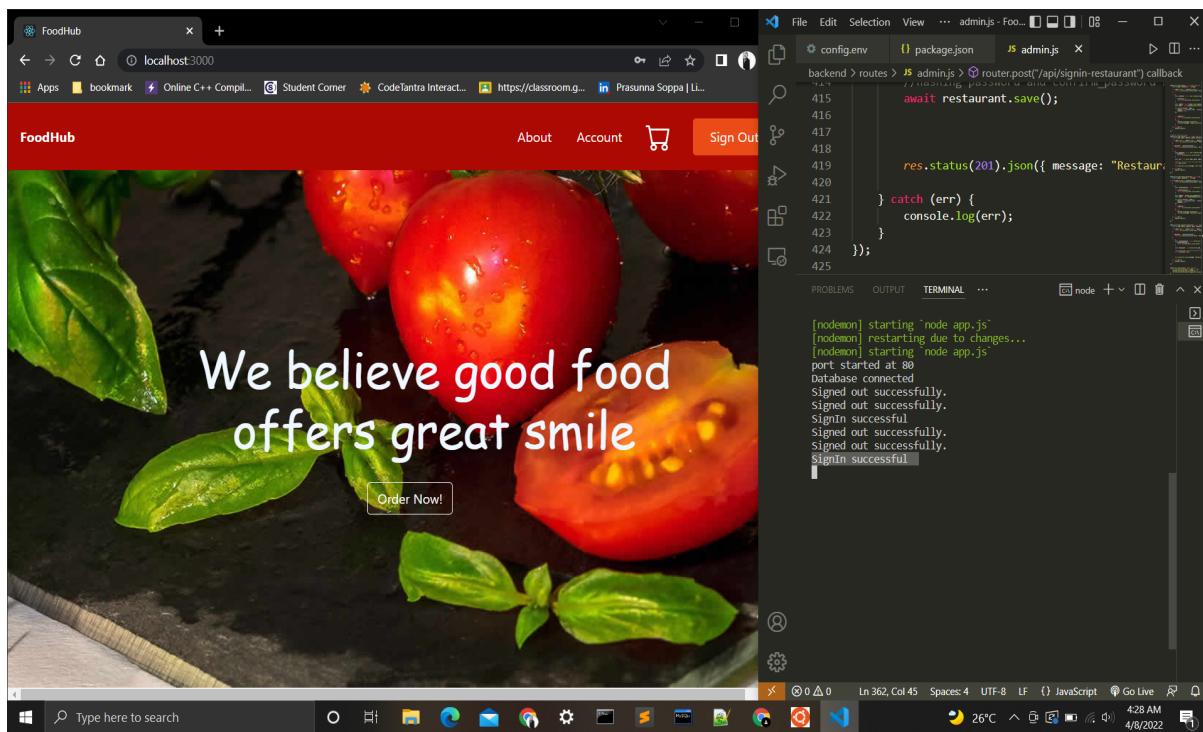
Input: Correct values are given

Expected Output: Successful Login.

Output:



Successful Login:

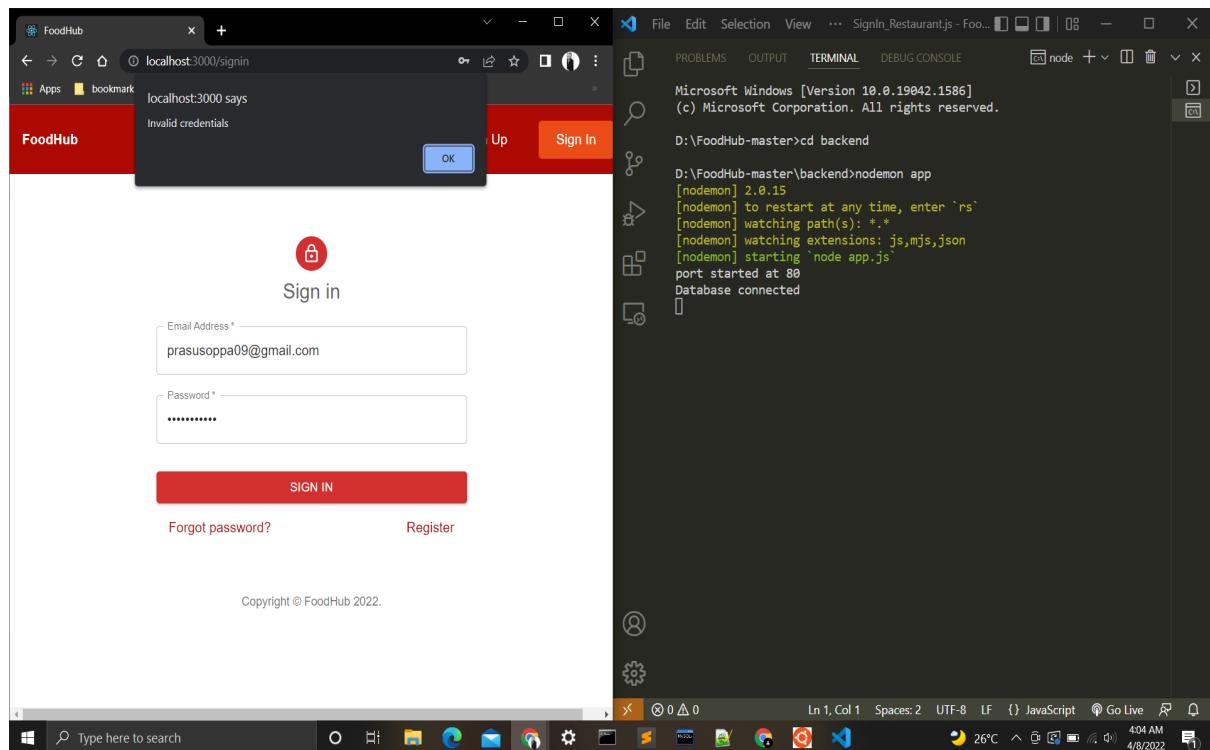


## Case 2: Incorrect input

Input: Incorrect credentials are given.

Expected Output: Invalid credentials (No Login).

Output:



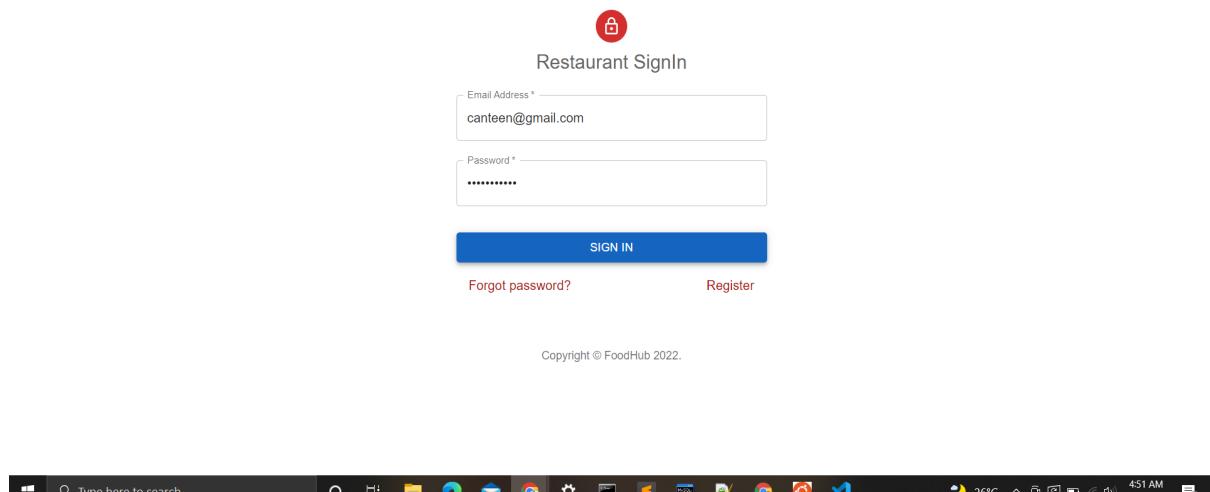
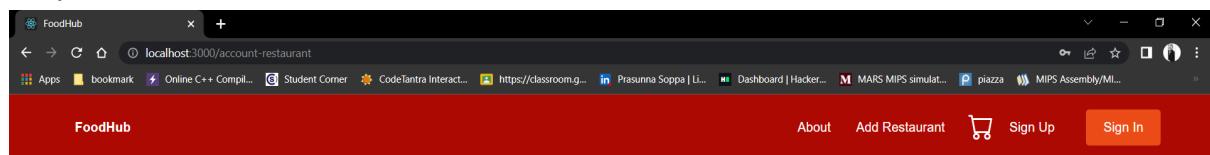
#### 4. Restaurants should enter correct credentials to login to their account:

##### **Case 1: Correct input**

Input: Correct values are given

Expected Output: Successful Login.

Output:



## Successful Login:

The screenshot shows a browser window for 'FoodHub' at [localhost:3000/account-restaurant](http://localhost:3000/account-restaurant). On the left, there's a sidebar with a photo of a restaurant interior and the text 'IIITG Canteen'. In the center, an 'Edit Profile' form is displayed with fields for Name ('IIITG Canteen'), Contact ('98732'), Email ('canteen@gmail.com'), and Address ('IIITG Academic Building'). A red 'UPDATE PROFILE' button is at the bottom. To the right, an 'Update Menu' form shows fields for Name ('Dish Name'), Price ('₹'), and Status ('Available'). A red 'SAVE' button is at the bottom. Below these forms is a 'Menu' table with columns 'Name', 'Price (₹)', and 'Status'. The terminal window on the right shows logs from node.js, including messages about starting the app, connecting to the database, and successful logins.

## Case 2: Incorrect input

Input: Incorrect credentials are given.

Expected Output: Invalid credentials (No Login).

Output:

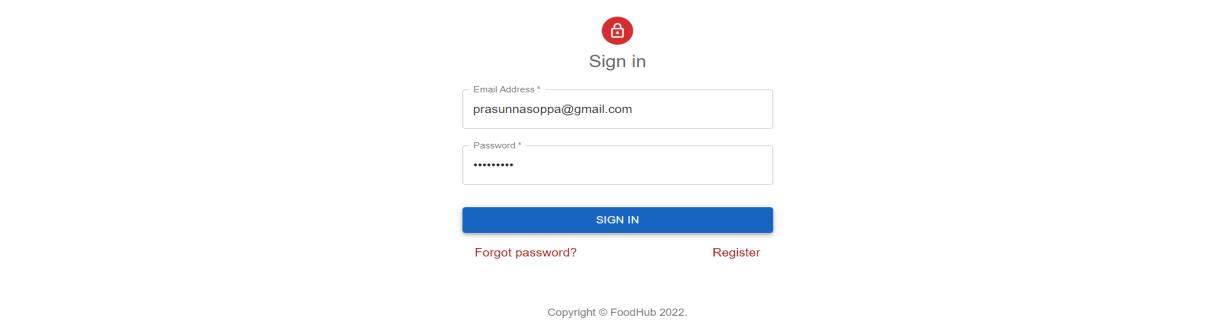
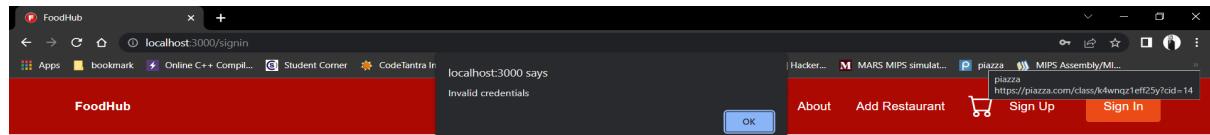
The screenshot shows a browser window for 'FoodHub' at [localhost:3000/signin-restaurant](http://localhost:3000/signin-restaurant). A modal dialog box is centered, displaying the message 'localhost:3000 says Invalid credentials' with an 'OK' button. The main page has a red header with the 'FoodHub' logo and navigation links for 'About', 'Add Restaurant', 'Sign Up', and 'Sign In'.

The screenshot shows the 'Restaurant SignIn' page. It features a lock icon and the title 'Restaurant SignIn'. There are two input fields: 'Email Address \*' containing 'canteen@gmail.com' and 'Password \*' containing '\*\*\*\*\*'. Below the fields are 'SIGN IN' and 'Forgot password?' buttons, and 'Register' text. At the bottom, it says 'Copyright © FoodHub 2022.'



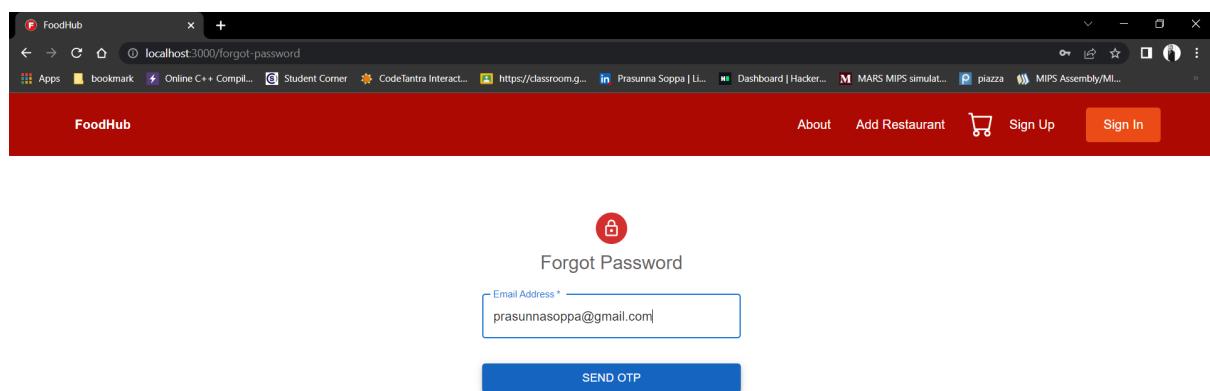
## 5. Users can change password through OTP Verification:-

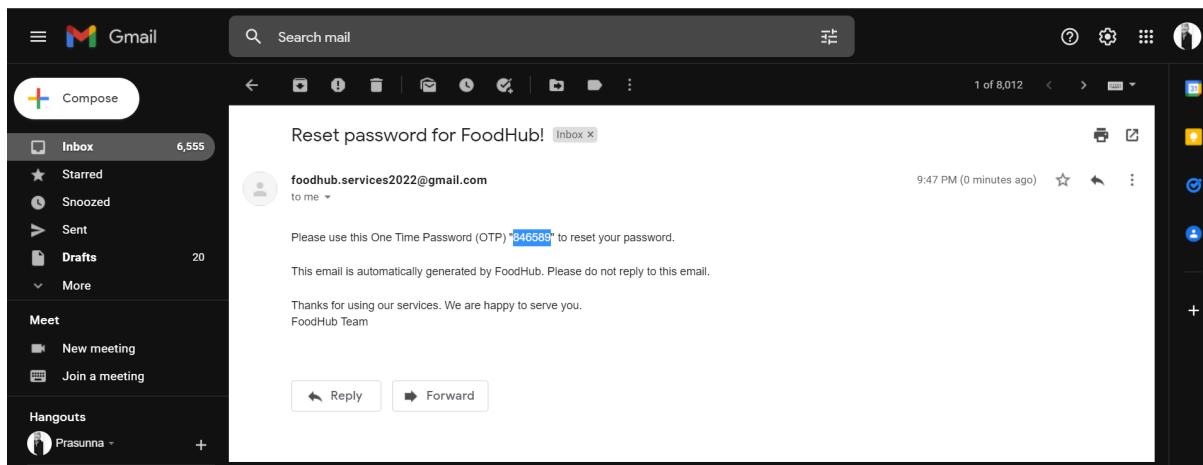
Before:



After clicking the button "Forgot Password", you will be asked for your mail id to which OTP is sent.

Verification:

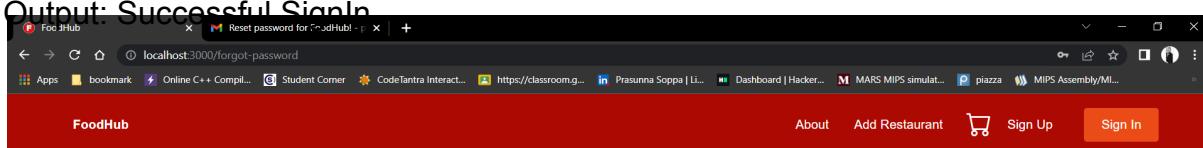




After:

Expected: Successful SignIn

Output: Successful SignIn



## Forgot Password

OTP \*

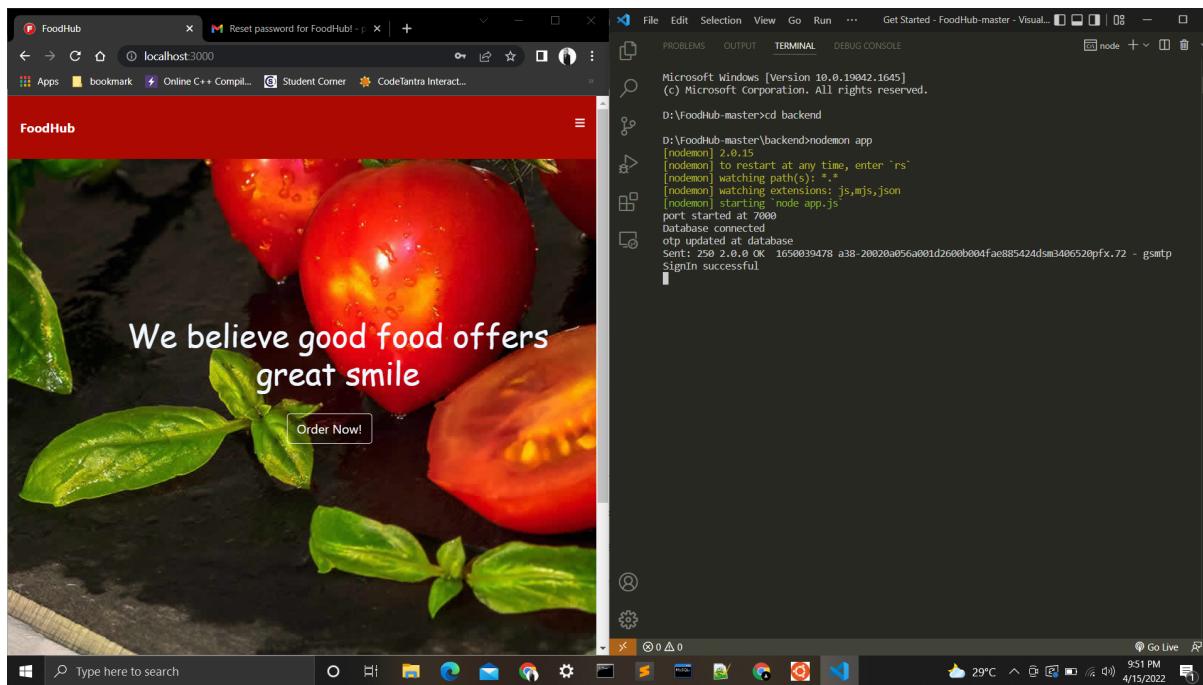
New Password \*

Confirm New Password \*

CHANGE PASSWORD

Copyright © FoodHub 2022.





## 6. Users can update phone no., email-id and address:-

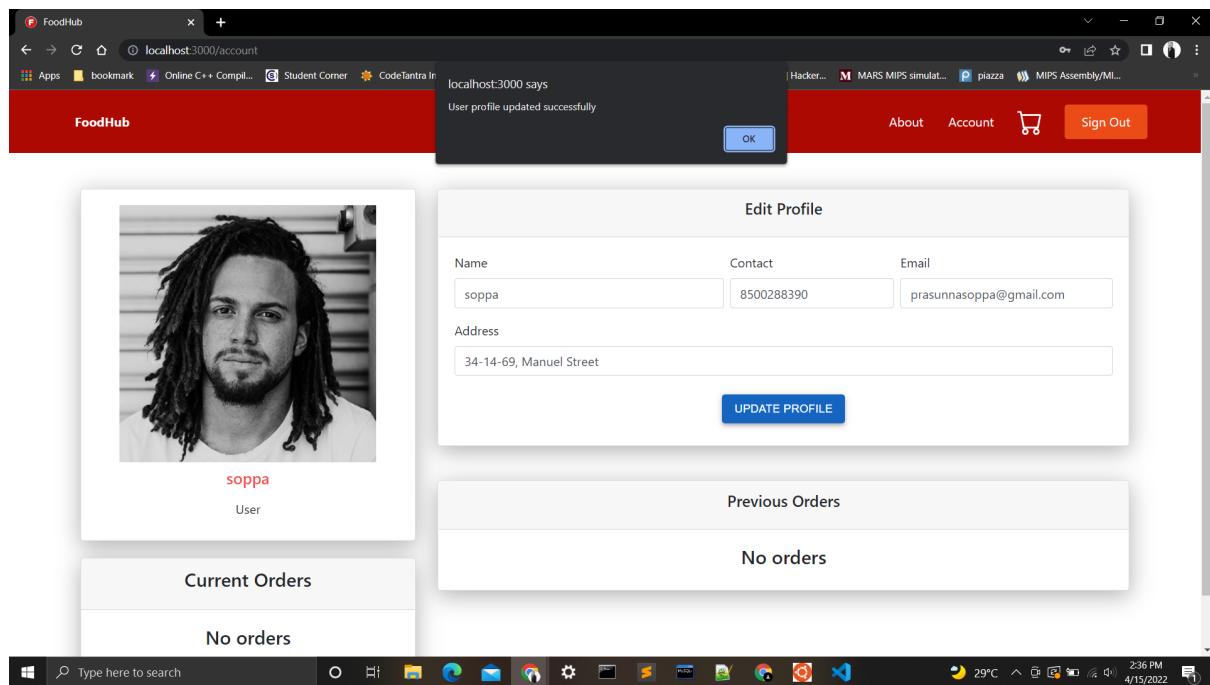
Before updation:

The screenshot shows the FoodHub account page at localhost:3000/account. At the top, there's a navigation bar with links for About, Account, a shopping cart icon, and Sign Out. The main area has a sidebar on the left with a user profile picture of a man with dreadlocks, the name "soppa", and the title "User". Below this is a section titled "Current Orders" with the message "No orders". To the right is a large "Edit Profile" form with fields for Name ("soppa"), Contact ("9491906885"), Email ("prasunna.soppa@gmail.com"), and Address ("34-14-69, Manuel Street"). A red "UPDATE PROFILE" button is at the bottom of this form. Below the profile form is a "Previous Orders" section with the message "No orders". The bottom of the screen shows a Windows taskbar with various pinned icons.

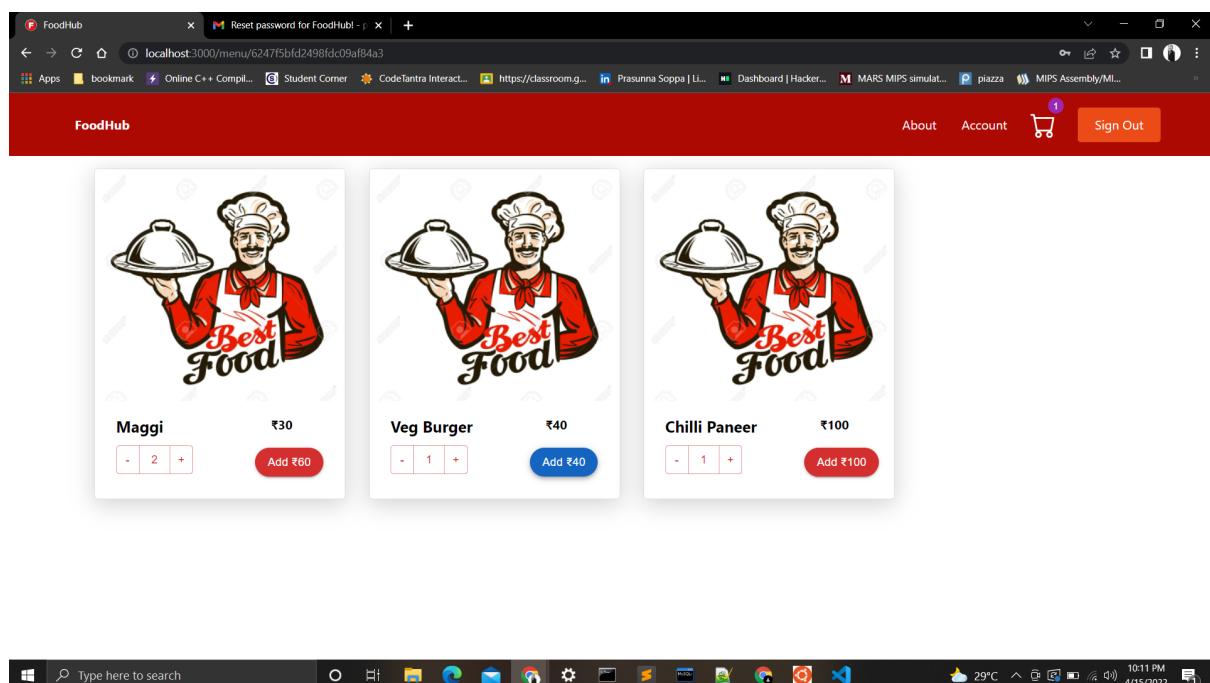
After updation:

Expected output: User profile updated successfully.

Result: User profile updated successfully.



## 7. Users can see the menus of respective food stalls and order from one stall at a time:-



After increasing the no.of dishes needed, click on the button “add” to add the food items to the cart.

Cart page:

**Cart**

Name	Restaurant	Quantity	Price	Amount
Maggi	Nescafe	2	30	60
Veg Burger	Nescafe	1	40	40
Chilli Paneer	Nescafe	1	100	100

Total Amount to be paid: ₹ 200

**CLEAR** **CHECKOUT**



Click on the checkout button to direct to the checkout page.  
Checkout page:

**Checkout**

① Delivery address — ② Payment details — ③ Review your order

**Delivery Address**

Address line 1 \*  
IIITG Acad Building

Address line 2

City \*  
Guwahati

Zip / Postal code \*  
781015

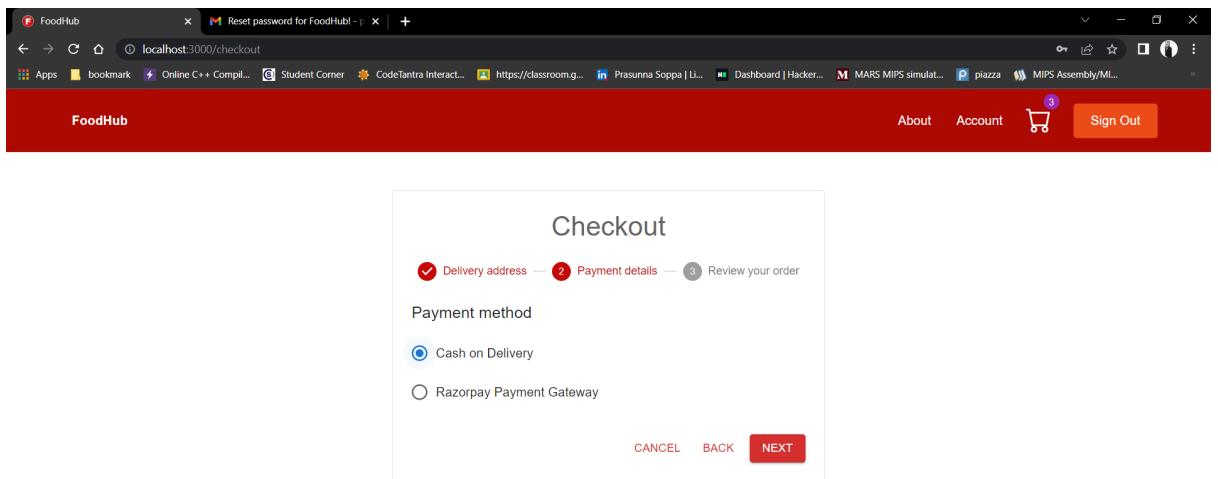
**CANCEL** **NEXT**



After filling the address to which food should be delivered and then click the next button.

You'll be directed to the payment for which you have two options:

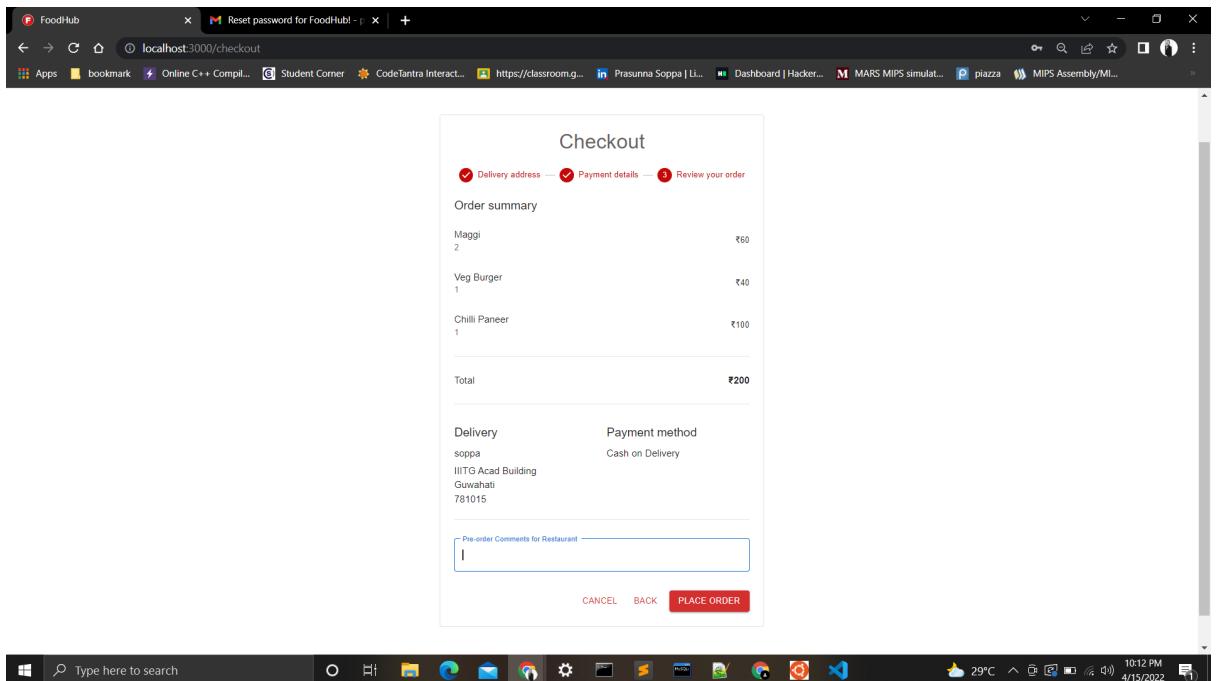
- i) Cash On delivery:-



Copyright © FoodHub 2022.

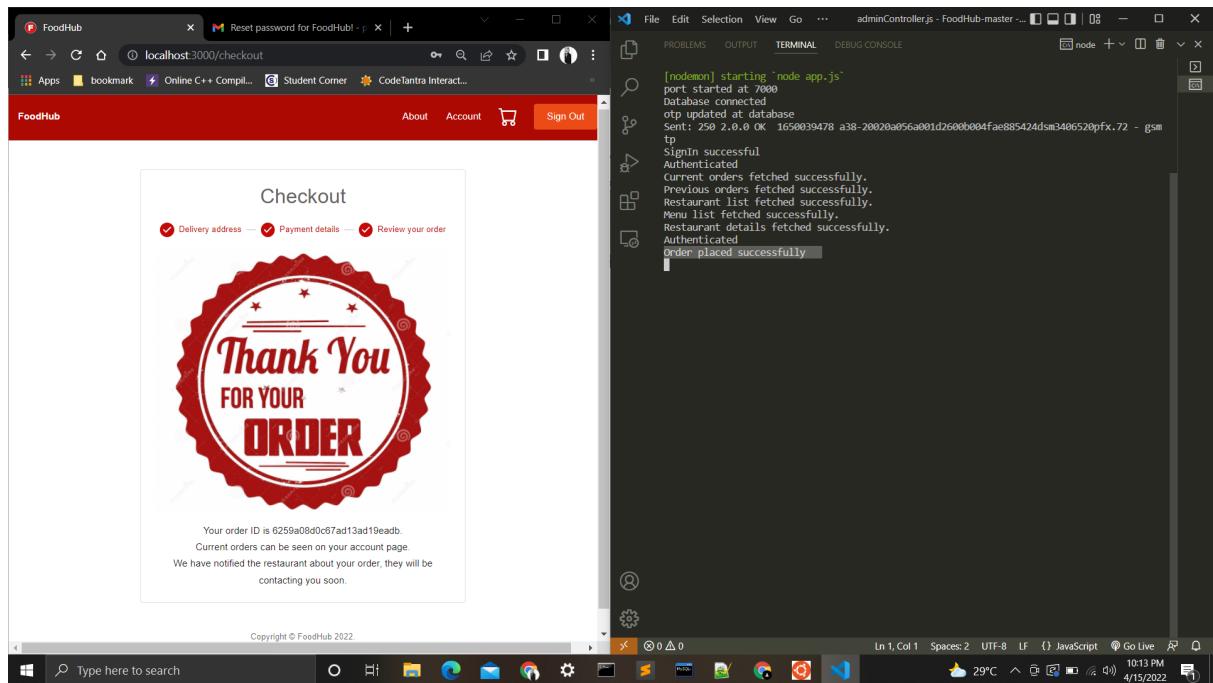


Click the next button to review your order and then place and can schedule your delivery by commenting in the preorder input box.

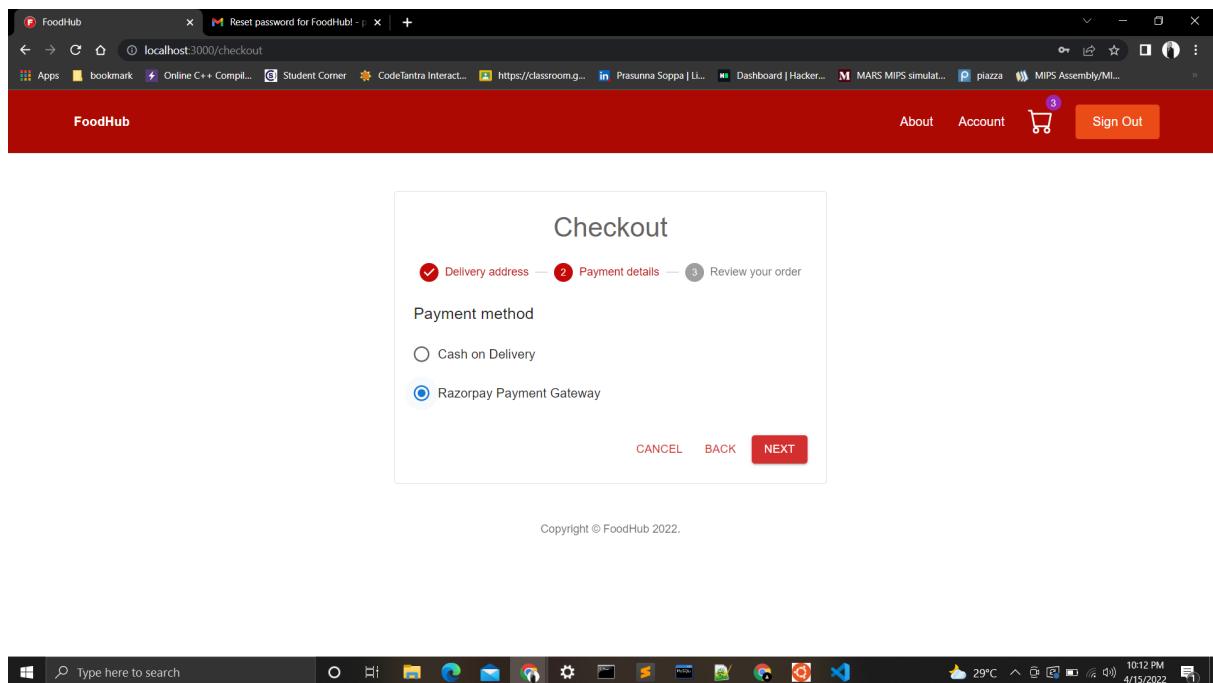


Click on the “Place Order” button to place your order.  
Expected o/p: Thank you for your order

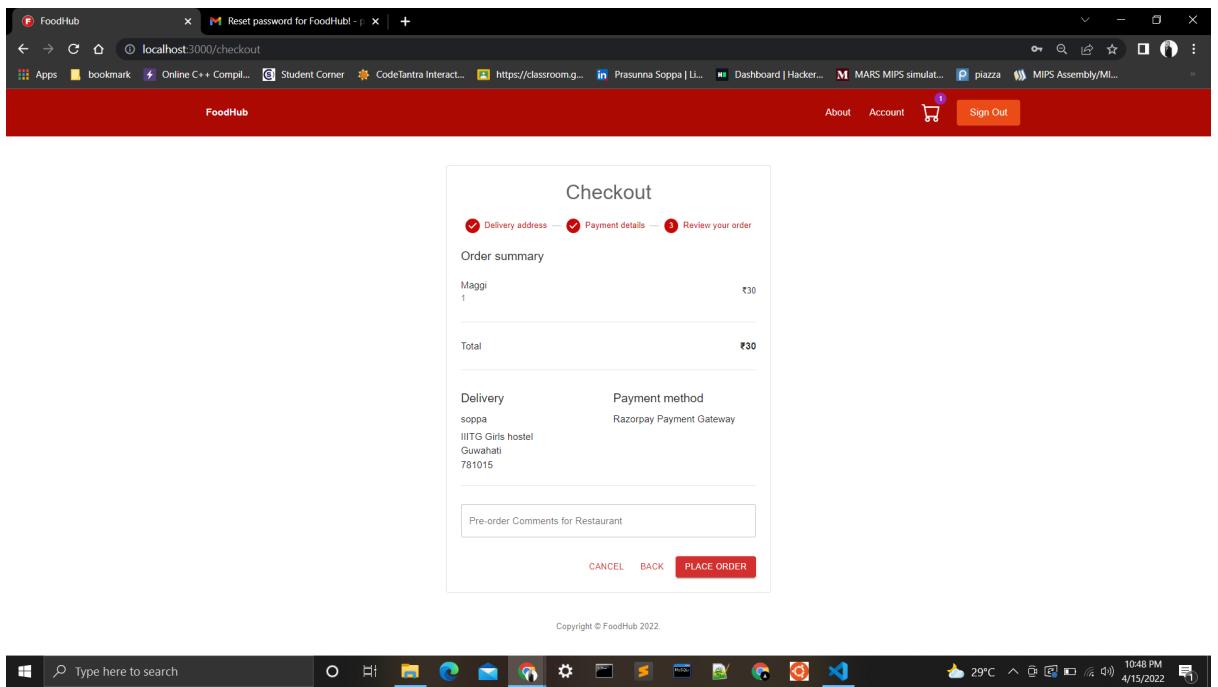
Output: Thank you for your order with order ID.



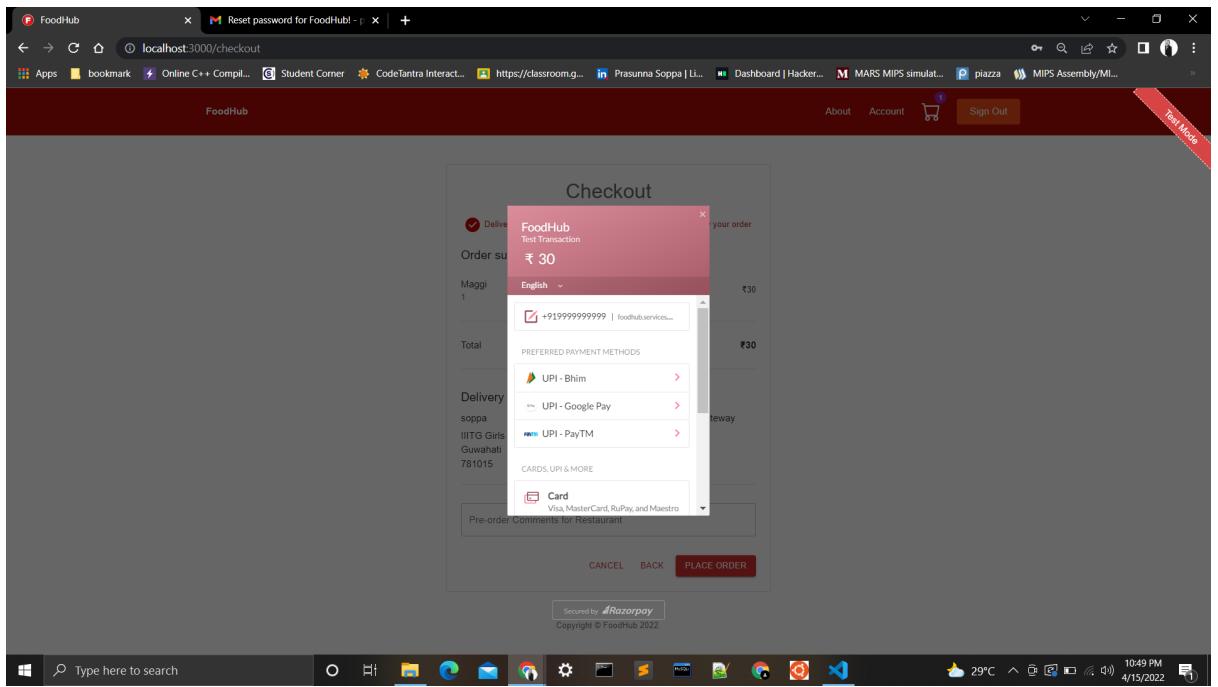
## ii) Razorpay Payment Gateway:-



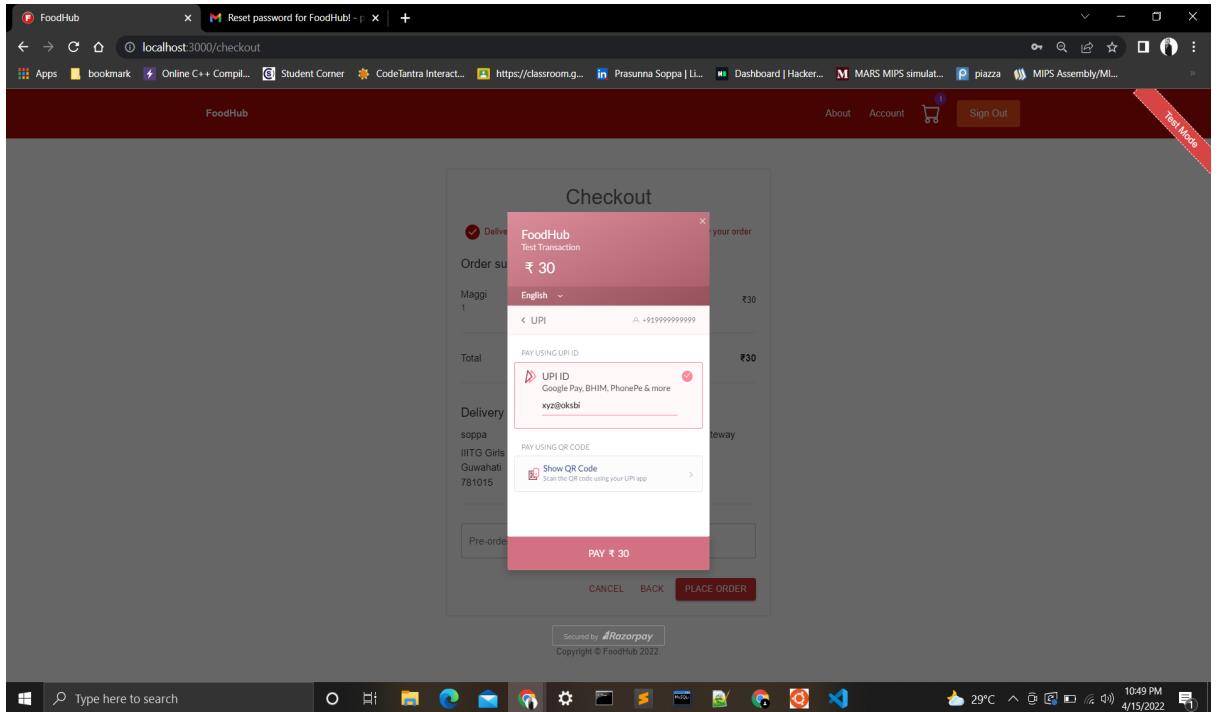
Click the next button to review your order and then place and can schedule your delivery by commenting in the preorder input box.



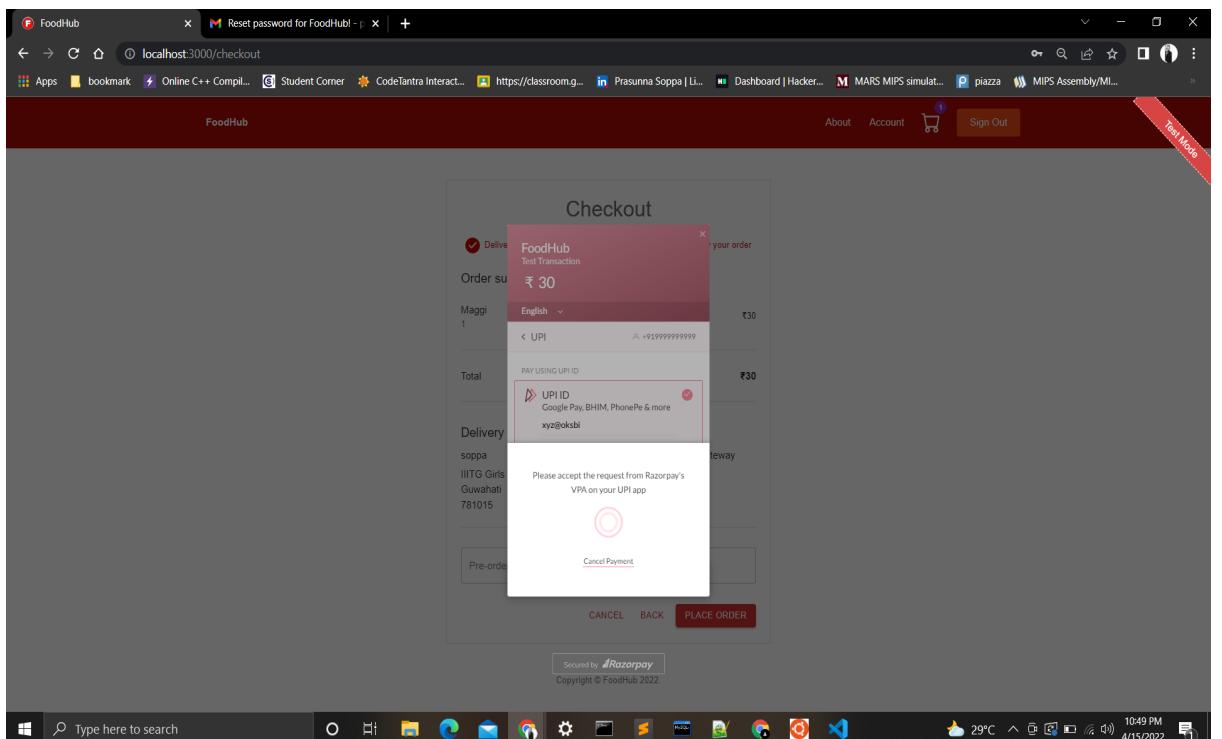
Click on the “Place Order” button to place your order.  
A box will pop up asking the method of transaction(convenient one).



Users should give the details of the account, so that a request will be sent for transaction.



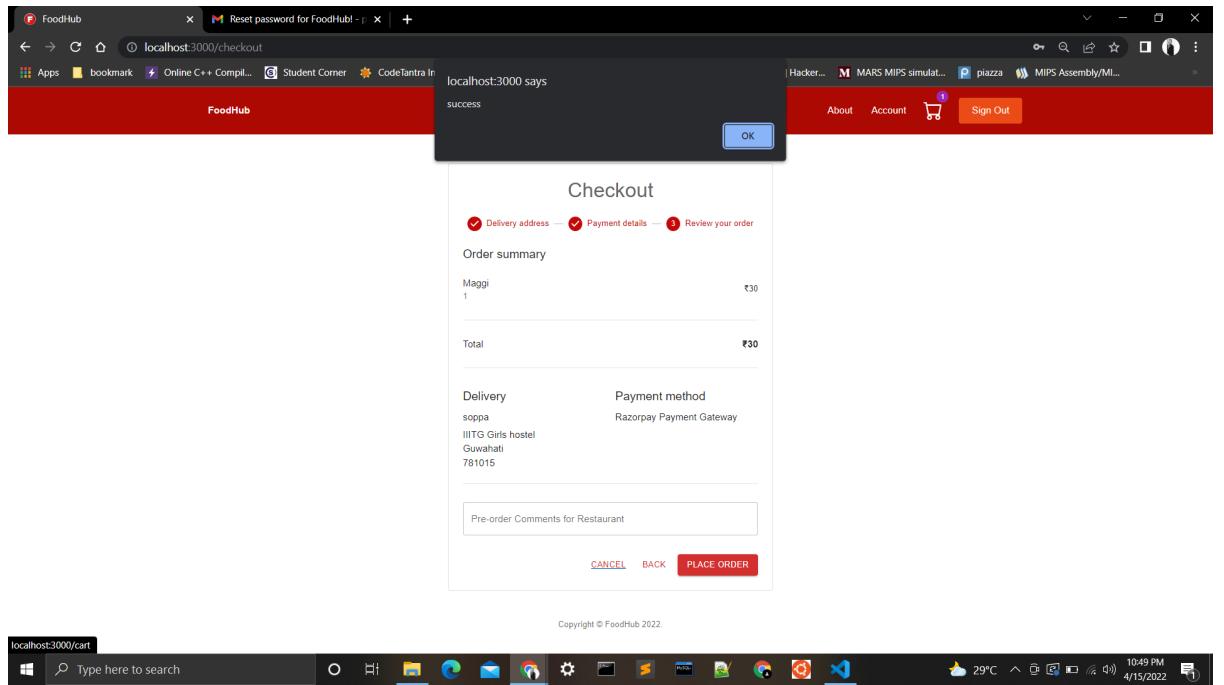
A request will be sent to the given account and should accept it for the transaction.



Once the request is accepted, payment will be done.

Expected o/p: Success

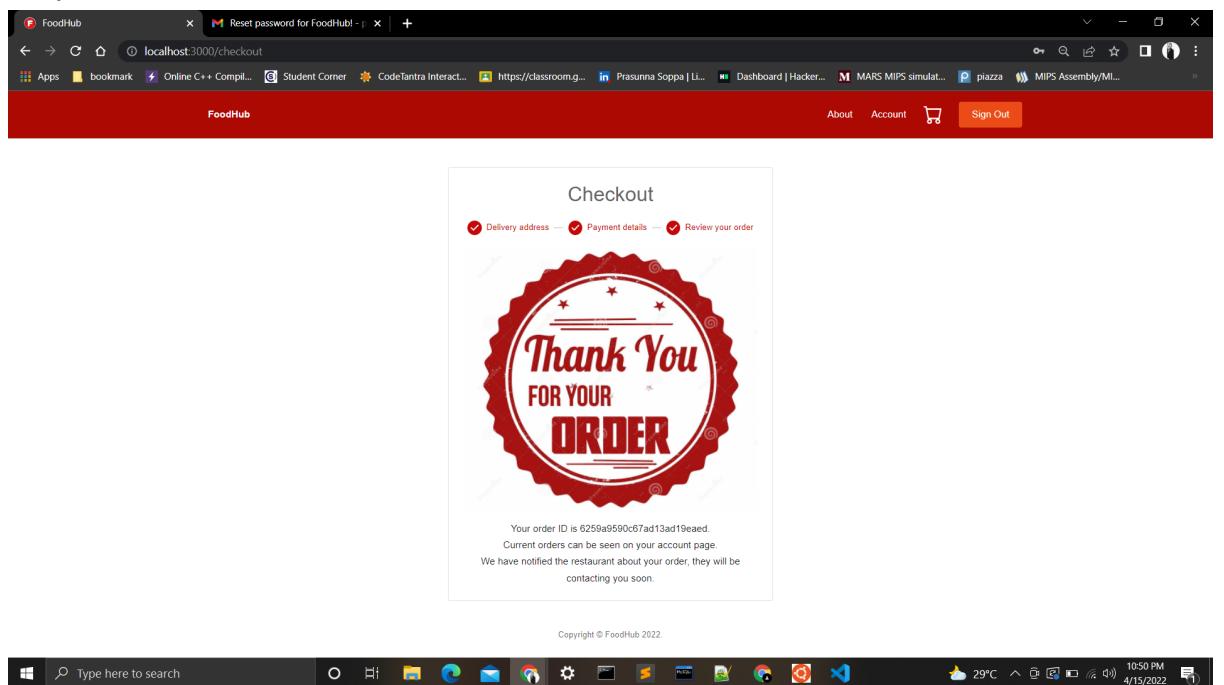
Output:



Click on the “Place Order” button to place your order.

Expected o/p: Thank you for your order

Output:



## 8.Restaurants can insert, update and remove the food items from the menu list:

### i)Inserting new dish:

Clicking on save, the dish will be added to the menu list.

The screenshot shows the FoodHub application interface. At the top, there's a header bar with 'FoodHub' and navigation links for 'About', 'Account', 'Orders', and 'Sign Out'. Below the header, there are two main sections: 'Edit Profile' and 'Update Menu'.

**Edit Profile:** This section contains fields for Name (Nescafe), Contact (9898781230), and Email (nescafe@gmail.com). There's also a field for Address (Bongora, IIITG) and a red 'UPDATE PROFILE' button.

**Update Menu:** This section has a form to add a new item. The 'Name' field is 'Veg Fried Rice', 'Price (₹)' is '75', and 'Status' is 'Available'. A red 'SAVE' button is at the bottom.

**Menu:** This section lists existing menu items with columns for Name, Price (₹), and Status.

Name	Price (₹)	Status
Maggi	30	Available
Veg Burger	40	Available
Chilli Paneer	100	Available
Masala Dosa	70	Not Available

### Before(from User side):

The screenshot shows the FoodHub application from a user's perspective. The menu items are displayed in three cards:

- Maggi**: ₹30. Quantity selector (0) and 'Add ₹0' button.
- Veg Burger**: ₹40. Quantity selector (0) and 'Add ₹0' button.
- Chilli Paneer**: ₹100. Quantity selector (0) and 'Add ₹0' button.



After adding the dish:

The screenshot shows the FoodHub website's menu page. At the top, there is a red header bar with the title "FoodHub" and navigation links for "About", "Account", and "Sign Out". Below the header, there are four menu items displayed in a grid:

Dish Name	Price (₹)	Action Buttons
Maggi	30	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/> <input type="button" value="Add ₹0"/>
Veg Burger	40	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/> <input type="button" value="Add ₹0"/>
Chilli Paneer	100	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/> <input type="button" value="Add ₹0"/>
Veg Fried Rice	75	<input type="button" value="-"/> <input type="button" value="0"/> <input type="button" value="+"/> <input type="button" value="Add ₹0"/>

ii) Updating the dish:

Dish's cost and its availability can be updated when needed.

Updating its cost:

Before:

The screenshot shows the FoodHub website's account page. At the top, there is a red header bar with the title "FoodHub" and navigation links for "About", "Account", "Orders", and "Sign Out". Below the header, there are two main sections: "Edit Profile" and "Update Menu".

**Edit Profile:**

Name	Contact	Email
Nescafe	9898781230	nescafe@gmail.com

**Address:**  
Bongora, IITG

**UPDATE PROFILE**

**Update Menu:**

Name	Price (₹)	Status
Maggi	30	Available
Veg Burger	40	Available
Chilli Paneer	100	Available
Masala Dosa	70	Not Available
Veg Fried Rice	75	Available

After clicking the save button, its cost will be updated.

The screenshot shows the FoodHub application running on a Windows operating system. The browser window has two tabs open: 'FoodHub' and 'localhost:3000/account-restaurant'. The main content area displays the 'Edit Profile' page for a user named 'Nescafe'. The profile includes a profile picture of a restaurant interior, the name 'Nescafe', contact number '9898781230', email 'nescafe@gmail.com', and address 'Bongora, IITG'. Below this is the 'Update Menu' section, which lists a single item: 'Maggi' at ₹45, marked as 'Available'. To the right is a 'Menu' table showing several items with their names, prices, and availability status.

Name	Price (₹)	Status
Maggi	45	Available
Veg Burger	40	Available
Chilli Paneer	100	Available
Masala Dosa	70	Not Available
Veg Fried Rice	75	Available

From User side:

The screenshot shows the FoodHub application running on a Windows operating system. The browser window has two tabs open: 'FoodHub' and 'localhost:3000/menu/6247f5bfd2498fd09a184a3'. The main content area displays a menu page for 'Best Food'. It features four items: 'Maggi' (₹45), 'Veg Burger' (₹40), 'Chilli Paneer' (₹100), and 'Veg Fried Rice' (₹75). Each item is accompanied by a small illustration of a chef holding a tray. Below each item is a quantity selector with buttons for minus, zero, and plus, and a red 'Add ₹' button.

Its availability can be updated by changing its status from available to non-available or the other way.

FoodHub

localhost:3000/account-restaurant

FoodHub

About Account Orders Sign Out

Nescafe

Edit Profile

Name Contact Email

Nescafe 9898781230 nescafe@gmail.com

Address

Bongora, IIITG

UPDATE PROFILE

Update Menu

Name

Maggi

Price ₹ Status

45 Available

SAVE

Menu

Name	Price ₹	Status
Maggi	45	Not Available
Veg Burger	40	Available
Chilli Paneer	100	Available
Masala Dosa	70	Not Available
Veg Fried Rice	75	Available

From User side:

Only available dishes are shown in the menu for the user.

A screenshot of a web browser displaying the FoodHub website. The page shows a menu with three items: "Veg Burger" (₹40), "Chilli Paneer" (₹100), and "Veg Fried Rice" (₹75). Each item features a cartoon chef holding a tray with a lid. Below each item is a quantity selector with buttons for minus, zero, plus, and "Add to Cart".

iii) Dish can be removed by just changing its status from available to non-available.

## 9.Restaurants can view the current orders and previous orders:

Current orders:

The screenshot shows the 'Current Orders' section of the FoodHub application. It lists two orders with the following details:

Dish	Quantity	Amount	Customer	Email	Contact	Delivery Address	Payment Method
Maggi	1	30	soppa	prasunnasoppa@gmail.com	8500288390	IITG Girls hostel Guwahati 781015	Razorpay Payment Gateway
Maggi	2	60	soppa	prasunnasoppa@gmail.com	8500288390	IITG Acad Building Guwahati 781015	Cash on Delivery
Veg Burger	1	40	soppa	prasunnasoppa@gmail.com	8500288390	IITG Acad Building Guwahati 781015	Cash on Delivery

Once the food is delivered, that order will be moved to the “previous order” section on clicking the “received” button from the user account.

After clicking on the received button from user side:

The screenshot shows the 'Previous Orders' section of the FoodHub application. It lists three orders with the following details:

Dish	Quantity	Amount	Customer	Email	Contact	Delivery Address	Payment Method
Maggi	2	60	soppa	prasunnasoppa@gmail.com	8500288390	IITG Acad Building Guwahati 781015	Cash on Delivery
Veg Burger	1	40	soppa	prasunnasoppa@gmail.com	8500288390	IITG Acad Building Guwahati 781015	Cash on Delivery
Chilli Paneer	1	100	soppa	prasunnasoppa@gmail.com	8500288390	IITG Acad Building Guwahati 781015	Cash on Delivery

From User's side:

Before clicking on the “received” button:

The screenshot shows a web browser window for 'FoodHub' at 'localhost:3000/account'. The top navigation bar includes links for 'Dashboard | Hacker...', 'MARS MIPS simul...', 'piazza', and 'MIPS Assembly/MI...'. The main content area displays a user profile picture of a man with dreadlocks, labeled 'soppa' and 'User'. Below this is a 'Current Orders' section. It lists two orders under 'Nescafe':

- Maggi (Quantity: 1) with a red 'RECEIVED' button.
- Nescafe (Order ID: 6259a08d0c67ad13ad19eadb) with a blue 'RECEIVED' button.

A 'Previous Orders' section is shown with the message 'No orders'.

After clicking on the “received” button:

The screenshot shows the same 'FoodHub' application window after one order has been marked as received. The 'Current Orders' section now shows:

- Maggi (Quantity: 1) with a red 'RECEIVED' button.
- Nescafe (Order ID: 6259a08d0c67ad13ad19eadb) with a blue 'RECEIVED' button.

The 'Previous Orders' section now displays a table of completed orders:

Sr	Name	Quantity	Amount
1	Maggi	2	60
2	Veg Burger	1	40
3	Chilli Paneer	1	100

Total Amount: ₹ 200

## 10. Restaurants and users can Log Out once they receive their order:

User:

The screenshot shows a Windows desktop environment. On the left is a browser window titled "FoodHub" with the URL "localhost:3000/signin". It displays a "Sign in" form with fields for "Email Address" and "Password", and buttons for "SIGN IN", "Forgot password?", and "Register". On the right is a code editor window titled "adminController.js" showing JavaScript code for a controller. The code includes imports for bcryptjs, dotenv, User, Restaurant, Order, and Menu models, and nodemailer for sending emails. A message in the terminal indicates an order was placed successfully and the user signed out successfully. The taskbar at the bottom shows various application icons.

Restaurant:

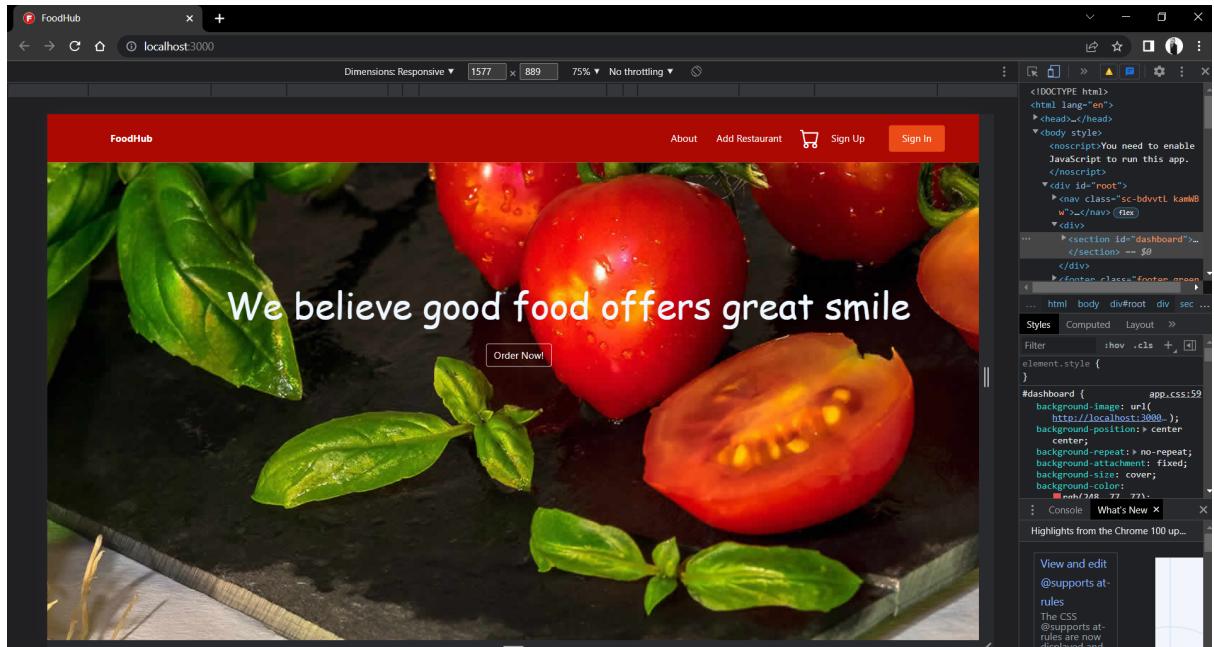
The screenshot shows a Windows desktop environment. On the left is a browser window titled "FoodHub" with the URL "localhost:3000/signin-restaurant". It displays a "Restaurant Signin" form with fields for "Email Address" and "Password", and buttons for "SIGN IN", "Forgot password?", and "Register". On the right is a code editor window titled "adminController.js" showing JavaScript code for a controller. The code includes imports for bcryptjs, dotenv, User, Restaurant, Order, and Menu models, and nodemailer for sending emails. A message in the terminal indicates the order was placed successfully and the user signed out successfully. The taskbar at the bottom shows various application icons.

## Non-Functional Requirements:-

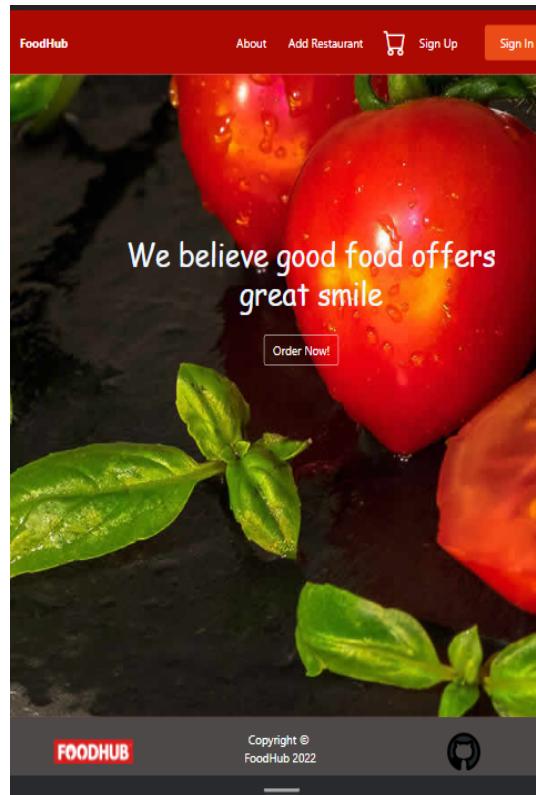
### ❖ Responsive and User friendly:

The application can be used on different platforms like different versions of OS/Mobiles, since it is responsive.

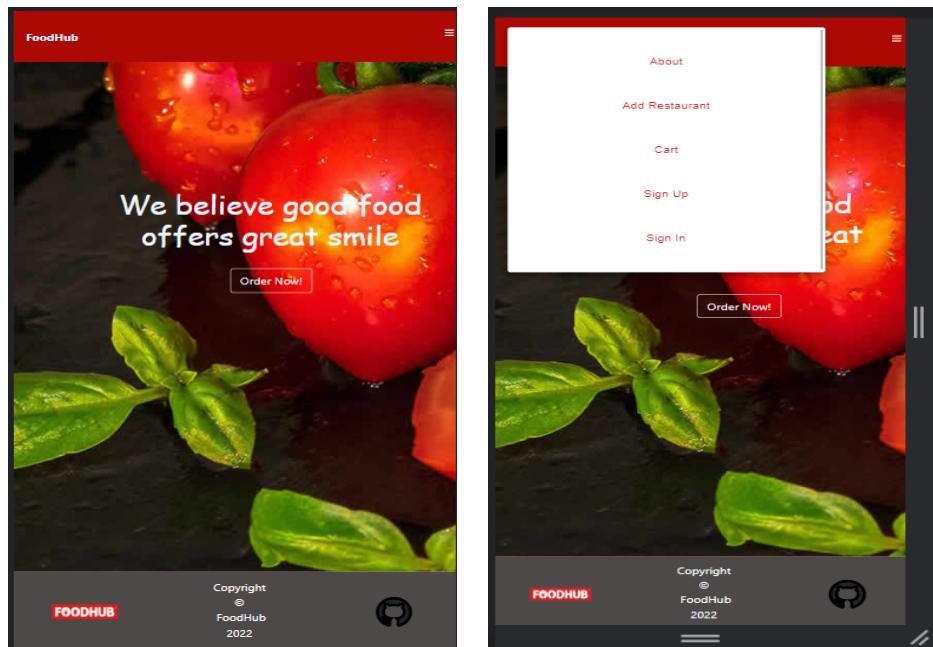
### Desktop View:



### iPad / NotePad View:



## Mobile view:



## ❖ Privacy and Security:

Personal data of the system is not disclosed to anyone.

During registration for users and restaurants, passwords are converted to hashed codes by using bcrypt-js encryption and are then stored in the database.

## For Users:

A screenshot of a MongoDB interface. On the left, there is a sidebar with a tree view showing a single node labeled "foodhub" which has children "menus", "orders", "restaurants", and "users". The main area is titled "Find" and shows a search bar with the placeholder "FILTER { field: 'value' }" and a "OPTIONS" button. Below the search bar, there is a "QUERY RESULTS 1-6 OF 6" section. It displays a single document with the following data:

```
_id:ObjectId("6247fd35d2498fdc09ef84b8")
name:"Swara"
address:"Bongora"
mobile:9812345609
email:"swara@gmail.com"
password:$2a$12$zj14gP6Gukh/Hf43.h6Qt4OZU2eF3DTwvRv1fgCjHn09Y7zjam4zG"
confirm_password:$2a$12$zj14gP6Gukh/Hf43.h6Qt4OZU2eF3DTwvRv1fgCjHn09Y7zjam4zG"
__v: 0
```

## For Restaurants:

The screenshot shows a MongoDB query interface. On the left, there's a sidebar with a tree view under the 'foodhub' collection, showing 'menus', 'orders', 'restaurants' (which is selected), and 'users'. At the top, there are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. Below the tabs is a 'FILTER' button with the expression '{ field: 'value' }'. To the right of the filter are 'OPTIONS', 'Apply', and 'Reset' buttons. At the bottom, it says 'QUERY RESULTS 1-1 OF 1' and shows a single document:

```
_id: ObjectId("6247f5bfd2498fdc09af84a3")
name: "Nescafe"
address: "Bongora, IIIIT"
mobile: 9898781230
email: "nescafe@gmail.com"
password: "$2a$12$cVPUzVjzKq/bPGelXZnTu1CX80q8AMVTzeKN3yh1LUUnNa4E/es0m"
confirm_password: "$2a$12$7Q5gkXPIny1fy3Yw.iL6Hu0vWauk1a.VpYC8a1TexH546obQzIM1"
__v: 0
```

## Checking the validity of all functions in our project:

```
PASS tests/paymentController.test.js
PASS tests/adminController.test.js

Test Suites: 2 passed, 2 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        4.164 s, estimated 9 s
Ran all test suites.

Watch Usage
› Press f to run only failed tests.
› Press o to only run tests related to changed files.
› Press p to filter by a filename regex pattern.
› Press t to filter by a test name regex pattern.
› Press q to quit watch mode.
› Press Enter to trigger a test run.
```

This is all the testing regarding our project.