

**EAT-SMART**

By

BIKASH GUPTA (2147113)

AAKASH SINGH (2147201)

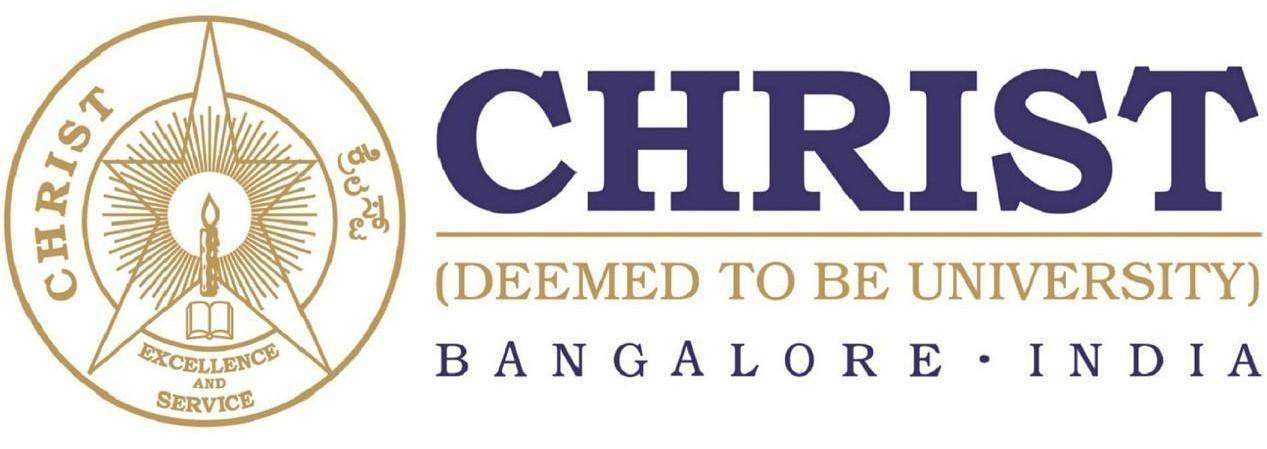
SUBHADIP SAHA (2147233)

Under the guidance of

Dr. Suresh K

A Project report submitted in partial fulfillment of the requirements for the award of degree of Master of Computer Applications of CHRIST (Deemed to be University)

May - 2022



**CERTIFICATE**

This is to certify that the report titled **Eat Smart** is a bonafide record of work done by **Aakash Singh, Subhadip Saha and Bikash Gupta** of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of V semester MCA during the year 2022.

**Head of the Department Faculty In charge**

Valued-by: Names : Aakash Singh

Subhadip Saha

Bikash Gupta

Register Number(s) : 2147201

2147233

2147113

Examination Centre : CHRIST (Deemed to be

University)

Date :

**ACKNOWLEDGEMENT**

During our project duration we have been accompanied and supported by many people. It is a pleasant aspect that we now have the opportunity to express our gratitude for all of them whose constant guidance and encouragement made all our efforts meet with success. First and foremost is Almighty God without whom nothing would have been possible, no words can articulate our deepest gratitude to you Lord for the inspiration and guidance which we have received.

We express our deep sense of gratitude to **Rev. Fr. Dr. Abraham. V. M**, Vice Chancellor of CHRIST (Deemed to be University) for providing the necessary facilities that helped us in accomplishing this work.

We feel a deep sense of gratitude to **Rev. Fr. Dr. Joseph C.C**, Pro Vice Chancellor of CHRIST (Deemed to be University) for his moral support.

We are deeply indebted to our Head of the department **Dr.Ashok Immanuel Sir,** Department of Computer Science, CHRIST (Deemed to be University), for rendering us invaluable support during the course of this project work.

We acknowledge our sincere thanks to **Dr. Shoney Sebastian**, Coordinator of MCA, Department of Computer Science, for his motivation and valuable guidance throughout the course of the project.

We are indebted to our project guide, **Dr.** **Suresh K,** Prof., Department of Computer Science, CHRIST (Deemed to be University), for guiding us with the true spirit throughout the project with profound sincerity and gratitude.

We would like to thank the faculty members of the Department of Computer Science, CHRIST (Deemed to be University) for providing us advice and motivation that helped us in staying on track.

# We consider it a pleasant duty to acknowledge the abundant help received from all our friends and classmates. We would also like to thank each and every one who contributed directly or indirectly towards the successful completion of our project work.

# ABSTRACT

People are increasingly relying on reviews from other people to decide which items to buy, which movies to watch, which books to read and where to eat.

Restaurants also need recommendation systems in terms of attracting more customers in the management side. In reality finding the favorite food and famous food especially in a new area is a challenging task. One solution to the problem is a recommender system that provides accurate and personalized recommendations. This will greatly reduce the effort and time needed to discover new restaurants. Eat-Smart is an online system to search restaurants. The system allows visitors to browse information about the restaurants, including searching restaurants., viewing/giving recommendations, and viewing/rating restaurants. Visitors can only view recommendations and view rating results.

The core of the application is a recommendation engine with an appropriate prediction model. The prediction models are built using restaurant dataset from kaggle, which contains detailed information of different restaurants in Lucknow. The output of the model may be recommending most popular restaurants and most popular food items served by the appropriate restaurant.

This project is based on Machine Learning algorithms using Python.

The second part of the application is a web application built around the recommendation engine using the newest web technology and framework available. The application provides personalized and relevant recommendations to users with high prediction accuracy.

**Table Of Contents**

**Sl No.                                       Topics                                  PageNo.**

1. **Introduction 7-10**

1.1 Project Overview -------------------------------------- 7

1.2 Existing System and Disadvantages------------------ 7-8

1.3 Proposed System and Advantages-------------------- 8

1.4 Problem Description—--------------------------------- 8-9

1.5 Literature Review—------------------------------------ 9-10

**2. Requirement Specification 11-13**

2.1 Functional requirements--------------------------------- 11

2.2 Non-Functional requirements--------------------------- 11-13

**3. Implementation 13-16**

3.1 Initial Design--------------------------------------------- 13

3.2 Input Design----------------------------------------------- 13-14

3.3 Output Design--------------------------------------------- 14

3.4 DFD--------------------------------------------------------- 15-16

3.5 Solution Architecture------------------------------------- 16

**4. Construction 17**

4.1 Software and Hardware Environments---------------- 17

**5. Screenshots 18-21**

**6. Conclusions & Learnings 22**

**List Of Figures**

| **Sl. No.** | **Fig Name** | **Page No.** |
| --- | --- | --- |
| 1 | DFD Level-0 | 16 |
| 2 | DFD Admin Level-1 | 16 |
| 3 | DFD User Level-1 | 17 |
| 4 | Solution Architecture | 17 |

# INTRODUCTION

EAT SMART is a restaurant recommendation system. It is an online system to search restaurants. Visitors can browse all restaurants in Bangalore, and get information about restaurant name, type, address, phone, rating, price, and map. The functions include searching restaurants, viewing/giving recommendations, and viewing/rating restaurants.

Additionally, the user can edit their account information and check their recommendation records. The biggest difference between visitors and members is that visitors can only view recommendations and rating results. If visitors would like to give recommendations or rate restaurants, they have to log in or sign up.

## 1.1 Objective

The objective of Eat-Smart is to let people check great restaurants through the website. ‘They can get the restaurant information and price here, so that they don't need to call restaurants or check the yellow page for details. They can find great restaurants by people's rating or recommendation to get suggestions. Those are from people's experience and comparison. In the other way, people try new restaurants from others' suggestions, and the restaurant gets more customers. It is an encouragement for the great restaurants. This document aims to capture the system requirements and features particularly related to recommendations to be implemented in the web application.

## 1.2 Existing System

* Zomato- Customers use the platform to search and discover restaurants, read and write customer generated reviews and view and upload photos, order food delivery, book a table and make payments while dining-out at restaurants.
* Swiggy-

-No Minimum Order. Order in for yourself or for the group, with no restrictions on order value.

-Live Order Tracking. Know where your order is at all times, from the restaurant to your doorstep.

-Lightning-Fast Delivery. Experience Swiggy's superfast delivery for food delivered fresh & on time.

* DineOut-

DineOut offers a host of experience elevating benefits that include an assured 25% Off on every meal, a 1+1 on buffet at 5 Star restaurants across the country, 10% Extra Promo Cash, exclusive access to curated culinary events and zero convenience fee.

## 1.3 Proposed System and Advantages

* The purpose of the project is to let people get ideas about which restaurant will be great for them. This system can give people some suggestions; also you can get others' opinions from this site.
* Furthermore, you can find the best restaurants by viewing the ratings page, which gathers many members' experience and response. This system is designed for people to search the information you send, and respond to all those restaurants that matched the customers' request.
* This system is like a communication bulletin for people who love to eat. In this site, there are many ways to search restaurants and rates, including by zip code, by type, by keyword, by distance, by price, and by recommendation search. These will be easier for people to use.
* This type of recommendation is very useful when we do not have any past historical data about the particular user. It works based on the principle of popularity and in the current trend.

**1.4 Problem Description**

For those who travel and keep changing places very frequently it is very hectic and plus they get to experience very different types of environments, of which they do not have much knowledge about. In such a situation, food can be an important factor to decide how you rate your trips and also recommend it to the people. Food can also attract people around the world to try it out if it were to be the best. In such scenarios, we need to find the right place, at reasonable cost, to serve us the best possible way. So there are few questions that must be addressed, such as:

1. How many types of foods are available in the restaurant?

2. Which is the closest to me with a good rating?

3. How many "similar" restaurants are available nearby me?

4. Do "similar" restaurants cost more? If so, what specialty do they have?

Expectations from this recommender system is to get answers for the questions, and in such a way that it uncovers all the perspective of managing recommendations. It is sighted to show:

1. What types of restaurants are present in a particular area?

2. Where are the similar restaurants present based on a preference to particular food?

3. How do different restaurants rank with respect to my preferences?

**1.5 Literature Survey**

Various methods are present for the development of restaurant recommendation systems.  Many of  the existing systems and functioning are as follows.  In  this  recommender  system,  they  developed recommendations  based  on  preferences  of the user.  It was  motivated  by  the  observation  that  a  user’s preference  against  an  item  is  affected  by  different aspects discussed in reviews. They first explored the topic modeling to discover the hidden aspects from review text.   Finally, they  utilized regression models to detect the user-restaurant relationship.  They  described  the  restaurant  recommendation system was very popular service whose accuracy and sophistication  keeps  increasing  every  day.  They presented  a  personalized  location-based  restaurant recommendation  system  integrated  in  mobile technology.  It ubiquitously  studied  the  user’s behavioral pattern  of recommendation  systems  and proposed methods to rectify it.  In  this  Research,  they  described  the  restaurant recommendation  system  with  machine  learning algorithms. In order to find a good machine-learning model, they have tried several  collaborating filtering methods to predict ratings between restaurants and users.  The  methods  they  have  implemented  are Slope  One,  k-Nearest  Neighbors  algorithm,  and multiclass  SVM  classification.  Our  evaluation  shows that  the  multiclass  SVM  classification  method outperforms the other methods.   For rating  prediction, they compare  user-based and item-based collaborative filtering  algorithms. Finally, architecture  is  given  to  support  the  building  of  a real-time recommendation service.  In  this  Proposed  system,  they  had  used  SVM  to predict the restaurant based on the user location. By developing  a  recommendation  system  which  could help  a  user  to  decide  which  restaurant  one  should visit, the person can save a lot of his time, efforts and money  and  thus  have  a  great  experience  and satisfaction.   There  are  various  factors  based  on  which  a  user decides  on visiting  a  restaurant  like  the  type  of cuisine  of  the  restaurant,  the  location  of  the restaurant,  the  ambiance,  price  range,  popularity, ratings, etc. Such  information is collected  and  made available on sites such as Yelp and Zomato.   Using  well  rounded,  open  source  dataset  provided by  Yelp  which  provides  data  not  only  of  the restaurant reviews, but also user-level information on their  preferred  restaurants  the  aim  is  to  build  an efficient recommendation system  for the  Yelp  users in  the form of  a software  application and thus  help them  predict  whether  they  will  like  visiting  a restaurant  or  not  by  applying  machine  learning techniques and algorithms.  In  this  paper,  they  studied  that  the  consumers use online  reviews  for  a  variety  of  reasons.  For  many products/  services,  there  are  a  large  number  of reviews  which  makes  it  difficult  for  consumers  to decide  which  reviews  to  pay  attention  to.  Hence, previous research has suggested that  online reviews websites  can  provide  a  customized  review  sorting system for each individual consumer.   Consequently,  drawing  upon  five  consumer segments  as  well  as  10  restaurant  characteristics found  in  the  literature,  we  propose  a  content-filtering  recommender  system  that  evaluates individual online reviews and assigns a numeric score to  each  review  for  each  of  the  five  consumer segments. The  numeric  scores can  later  be  used to sort  online  reviews  for  individual  consumers according to their taste for restaurants.

# REQUIREMENT SPECIFICATION

The following section presents the complete set of functional and non-functional requirements identified for the project. Functional requirements are listed first, according to their relationship to the overall system, customers Administrator. The non-functional requirements that pertain to safety, security, the interface, human engineering, qualification, operation, maintenance and performance are subsequently presented. The functional requirements have been specified using a natural language description and as such, the reader is directed.

## 2.1 Functional requirements:

i) Registration: The customer will be able to register by providing the details like first name, last name, user id, password, Email, Phone No. After all the validations are performed, if user id does not exist then user is allowed to register otherwise registration fails.

ii) Login: This module asks for username and password allowing the user to access the system.

iii) Restaurant Details: Users will be able to search the details of different restaurants based on their location, cuisine, ratings, cost.

iv) Customer Feedback: The system will give customers the ability to give feedback for the restaurants or overall services. In the feedback screen there are multiple choice questions each having two options “Satisfactory” and “Unsatisfactory”. At the end there is a submit button.

v) Add/Edit/Delete: The system gives ability to the admin to add, edit and delete restaurants. Using this feature an admin can add details of different restaurants.

## 2.2 Nonfunctional Requirements :

i) Adaptability: There can be a change in the restaurants and information stored in the database about employees and inventory.

ii) Flexibility: If need arises in the future, software can be modified to change the requirements.

iii) Interoperability: The data is transferred from the customer’s end to the system and then the system recommends restaurants to each customer. This way data is transferred from one part of the system to another.

iv) Maintainability: Software can be easily repaired if a fault occurs.

v) Portability: Software can be easily installed on devices and would run smoothly according to the requirement.

vi) Robustness: Software must have checks to ensure that the system

recommends the correct restaurant and the emails, phone numbers added are all valid.

(USER/GUEST, MEMBER)

Member:

1. Sign up
2. Search restaurant
3. View map
4. View recommendation
5. View rate
6. Rate Restaurants
7. Edit account

Guest:

1. Sign up
2. Search restaurant
3. View map
4. View recommendation
5. View rate

ADMINISTRATOR:

1. Edit restaurant
2. Add restaurant
3. Delete restaurant
4. Delete member
5. Edit recommendation

## 3. IMPLEMENTATION:

Overall, the system that we want to build is a recommender system in the restaurant domain. In this study, we use a user-based collaborative filtering method. If the user wants to find a restaurant recommended by another user, then the system will search for the similarity of preferences of the target user and all existing users by calculating the similarity between users and the similarity of the user attributes. Then the system searches the neighbors who have the biggest similarity with the target user, so that restaurants that have been given a rating by neighbors will be recommended to target users who have not rated the restaurant.

### 3.1 Initial Design:

Initial design of the EAT SMART- Restaurant Recommendation System web application is all about designing user interface for operating and satisfying user requirements. For that purpose, we have investigated well- known technologies and identified HTML, CSS, Python, Django etc.

We gathered requirements for this application in different ways and found that Python, Django for coding and HTML and CSS for attractive styles.

### 3.2 Input Design:

Input design is the process of converting the user originated inputs to a computer format. The input design involves determining what the inputs are, how the data should be performed, how to validate data, how to minimize data entry and how to provide a multi user facility. The design for handling input specifies how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and includes specifying the means by which actions are taken.

A system user interacting through a system must be able to tell the system whether to accept input, produce a report or end processing. The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information. If the data going into the system is incorrect then processing and outputs will magnify this error.

All input data are validated in the order and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate tables in the database.

We have to keep in mind the following things to design the system

* What data to input?
* What medium to use?
* Methods for performing input validation and steps to follow when errors occur

Input requirement gathering was one of the major trivial process in web or android

application development. The project involves text inputs. The inputs can be entered through keyboard and mouse. The text input is gathered by forms with text boxes.

### 3.3 Output Design:

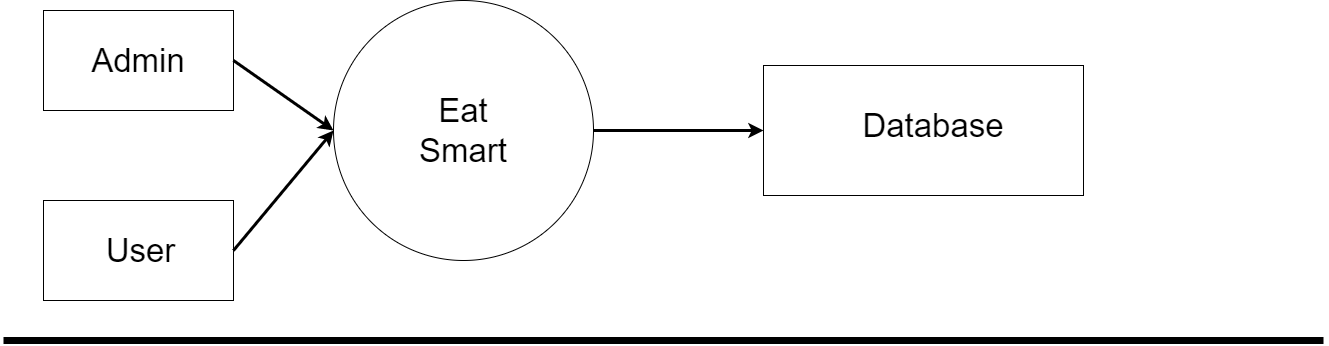
Effective output design will improve the clarity and performance of output. Output design phase of the system is concerned with the convergence of information to the end user friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making.

The Outputs from the EAT SMART- Restaurant Recommendation System consist of different formats. It provides search details in text format. The most attractive feature of this application is it shows the results of search contents after efficient filtering and pruning techniques. Efficient and eligible output design should improve the system’s relationship with the user and help in decision making.

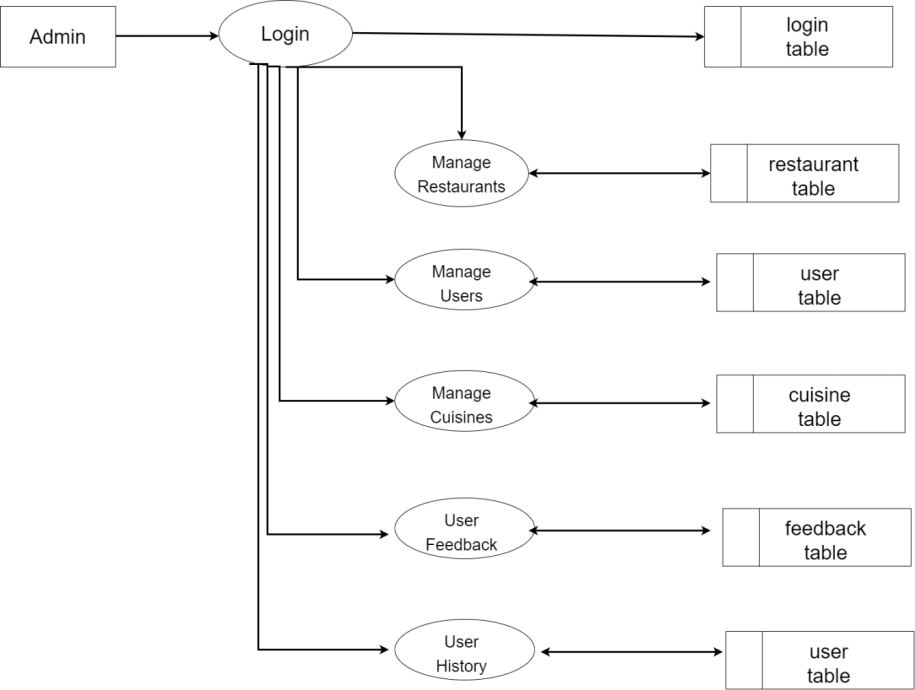
Outputs are the most important and direct source of information to the user and to the management. Output design generally deals with the results generated by the system. The output data is in the format of text.

### 3.4 Data Flow Diagram:

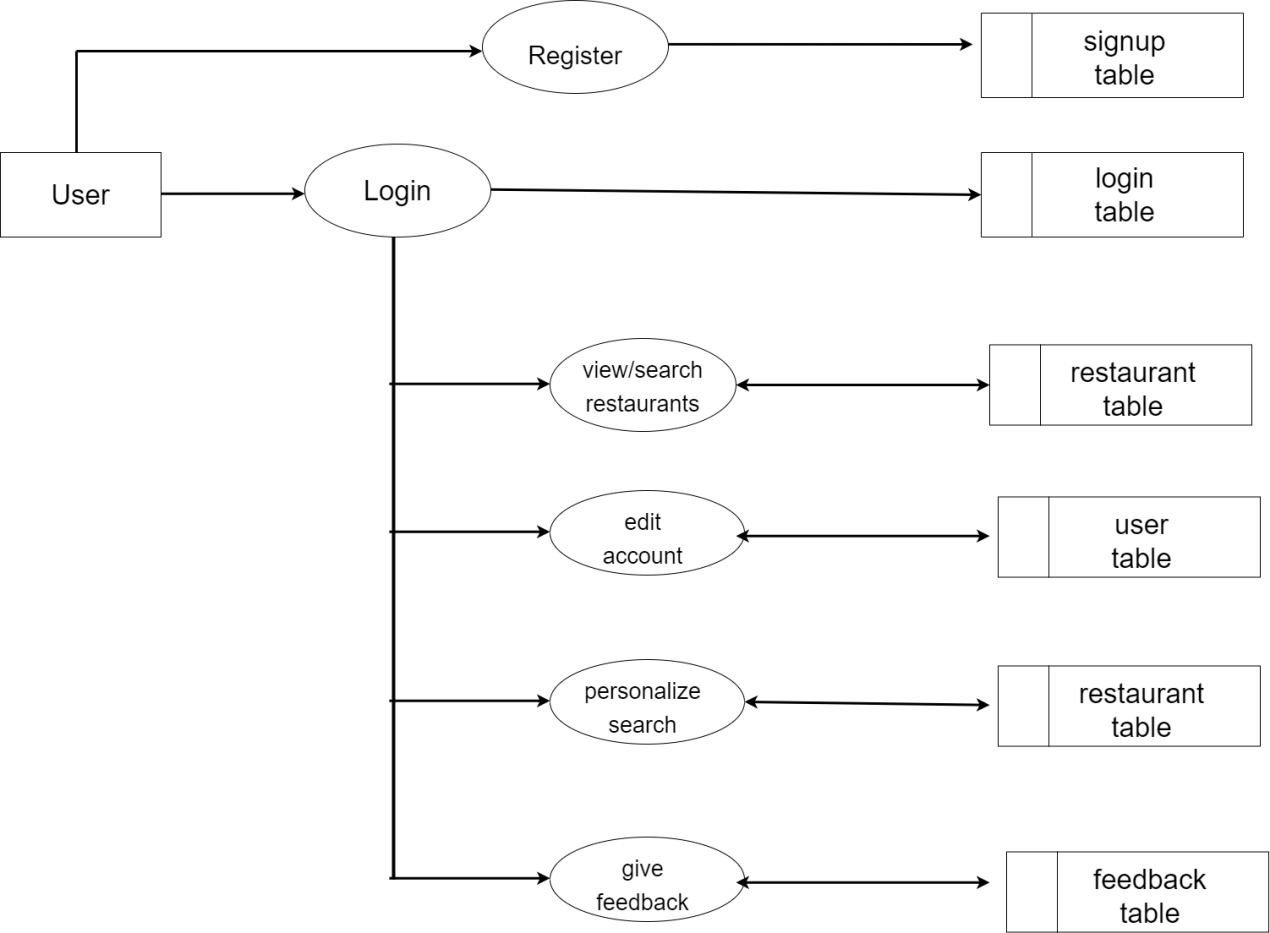
**3.4.1 LEVEL 0:**



### 3.4.2 Admin Level 1:

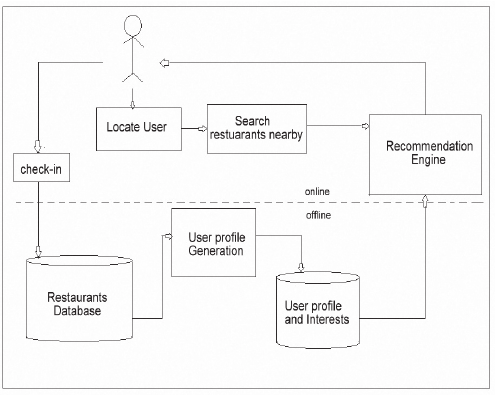


### 3.4.3 User Level 1:



### 

### 3.5 Solution Architecture:



# 

# 4. CONSTRUCTION

#### 4.1 Software Environments

Operating System: Windows 11/10/8/7 or XP

Frontend: HTML, CSS, JS

Dataset: Zomato Lucknow Dataset

Browser: Chrome/Mozilla Firefox/Microsoft Edge

#### 4.2 Hardware Environments

Processor: Intel core i3 or above

RAM: Minimum 4GB

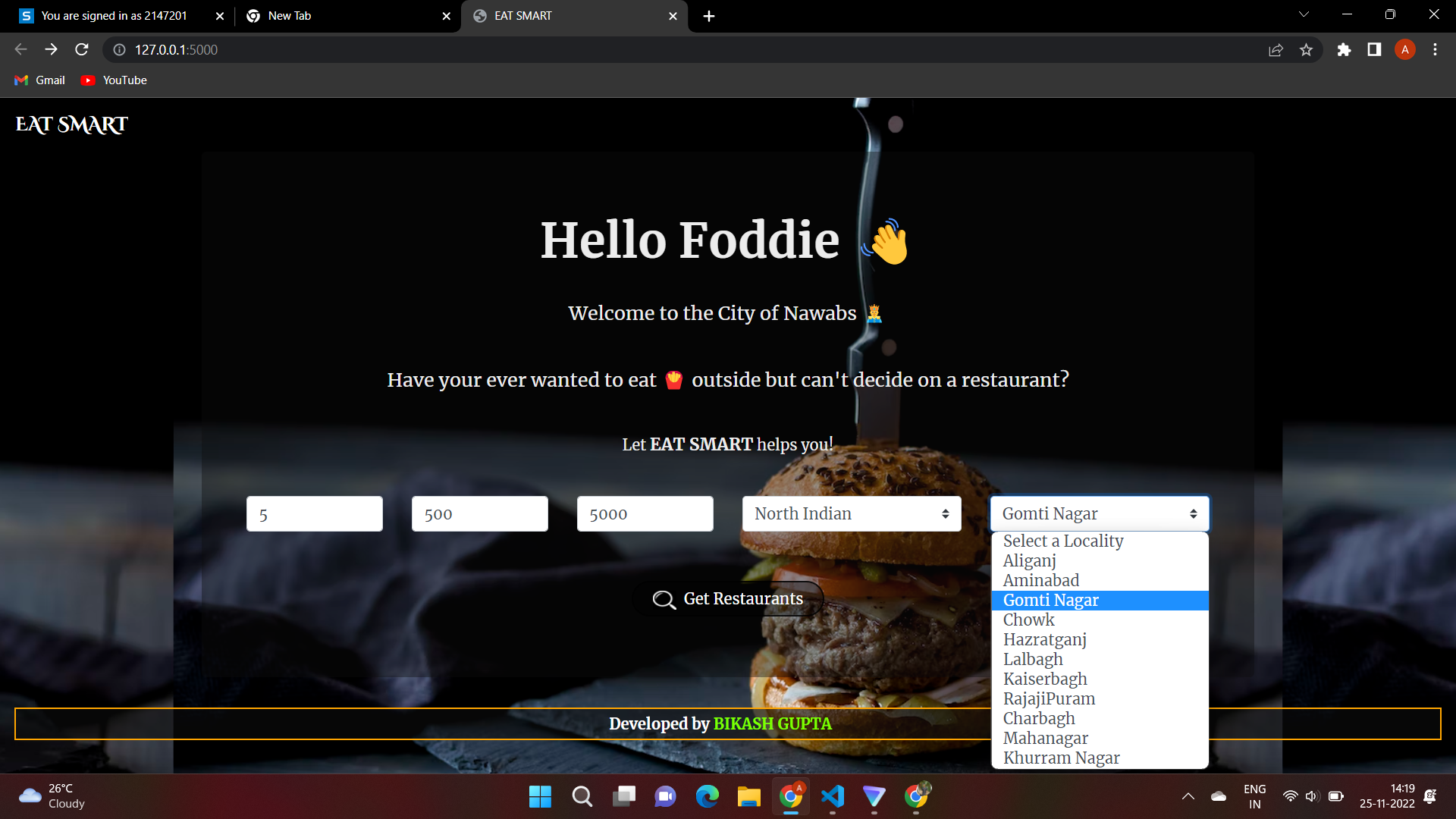
HDD: 1TB

# 5. Screen Shots

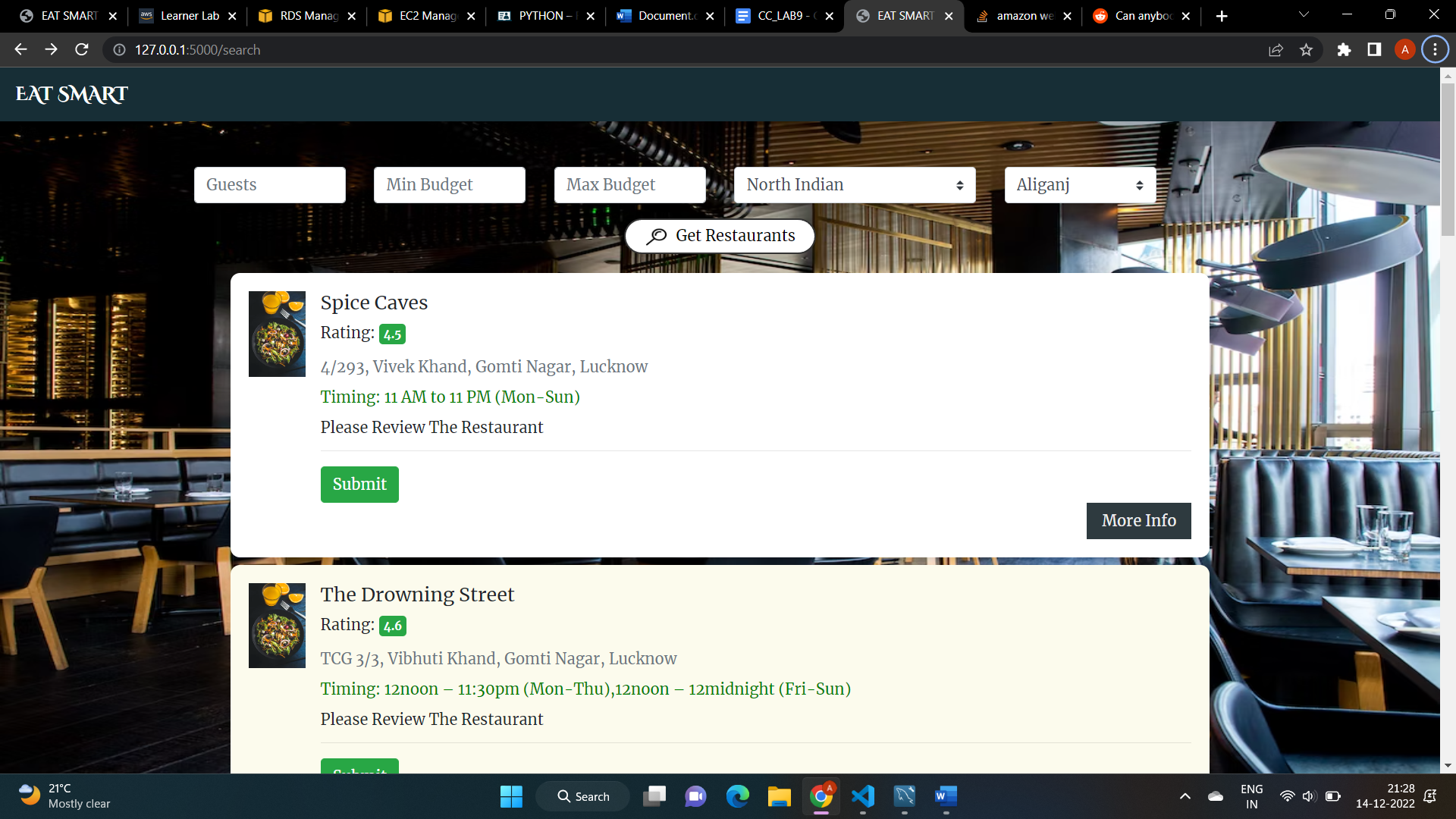
##### 5.1 Default landing page:

****

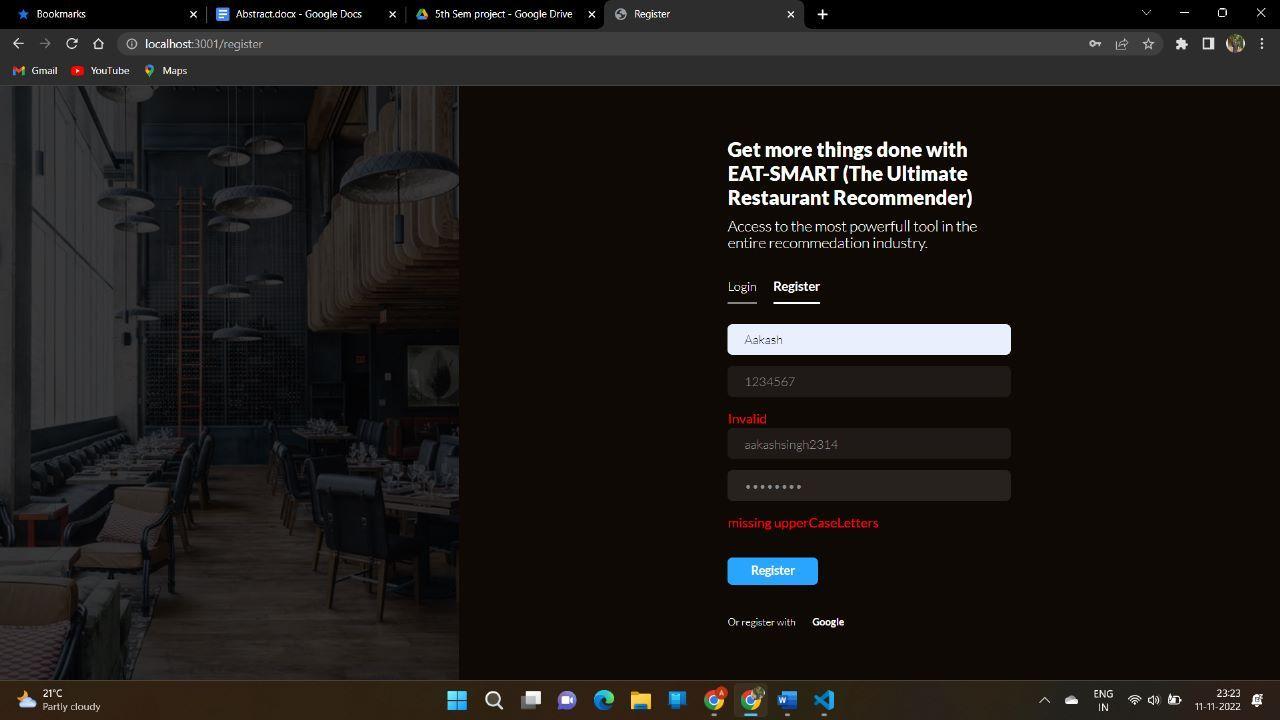
##### 5.2 Search page :

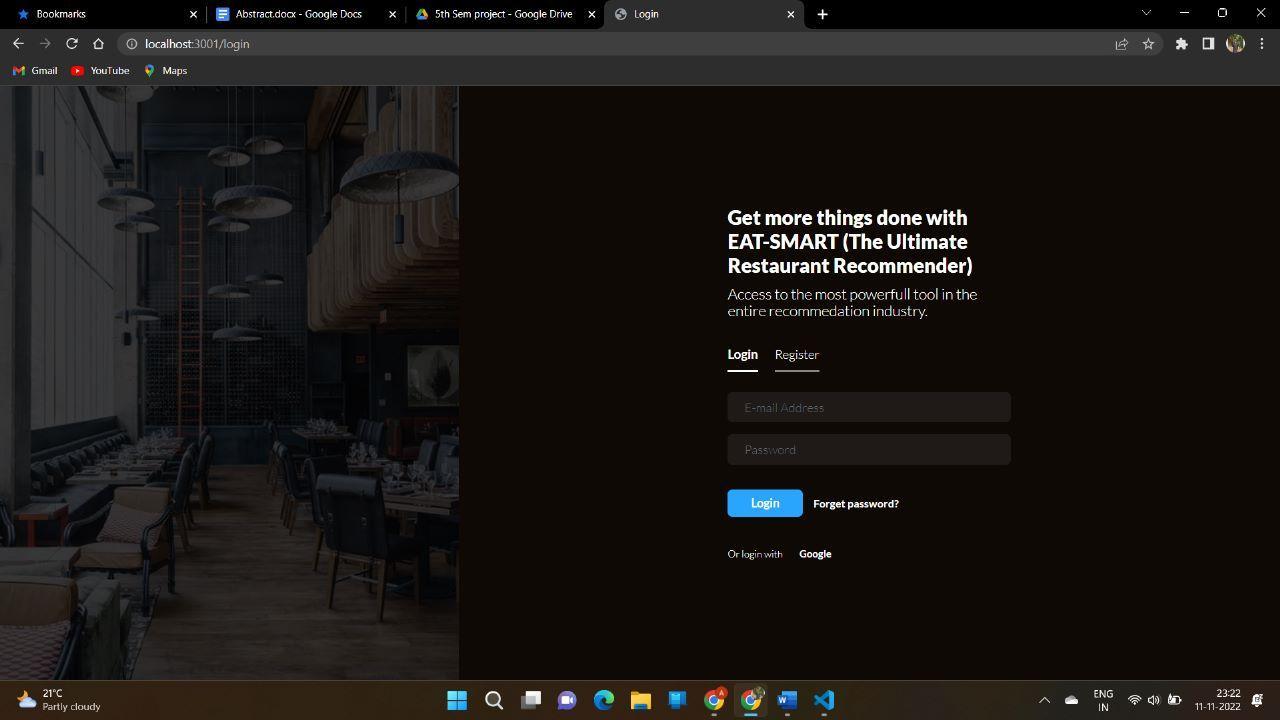


##### 5.3 Search result Recommendation page:

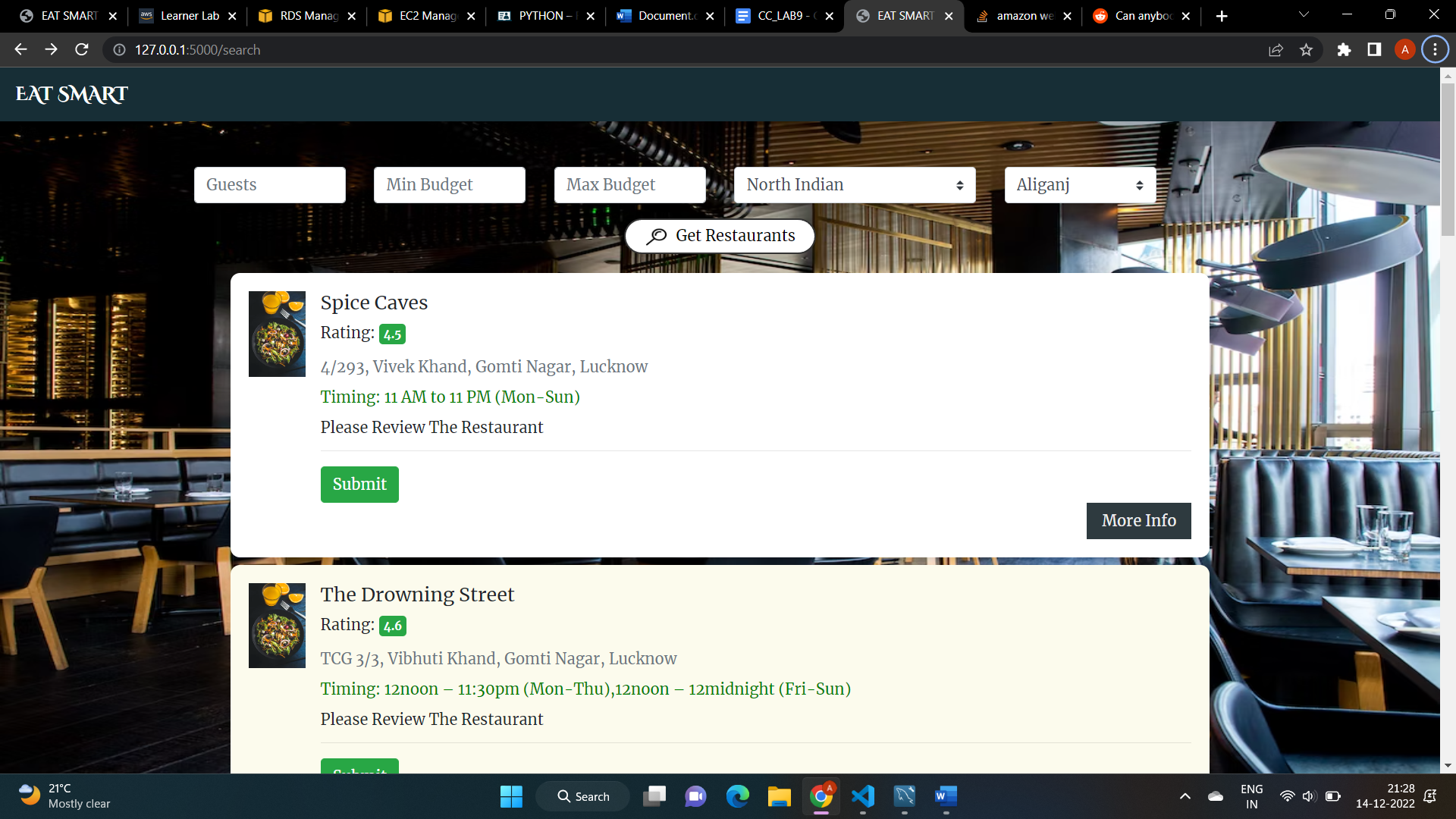
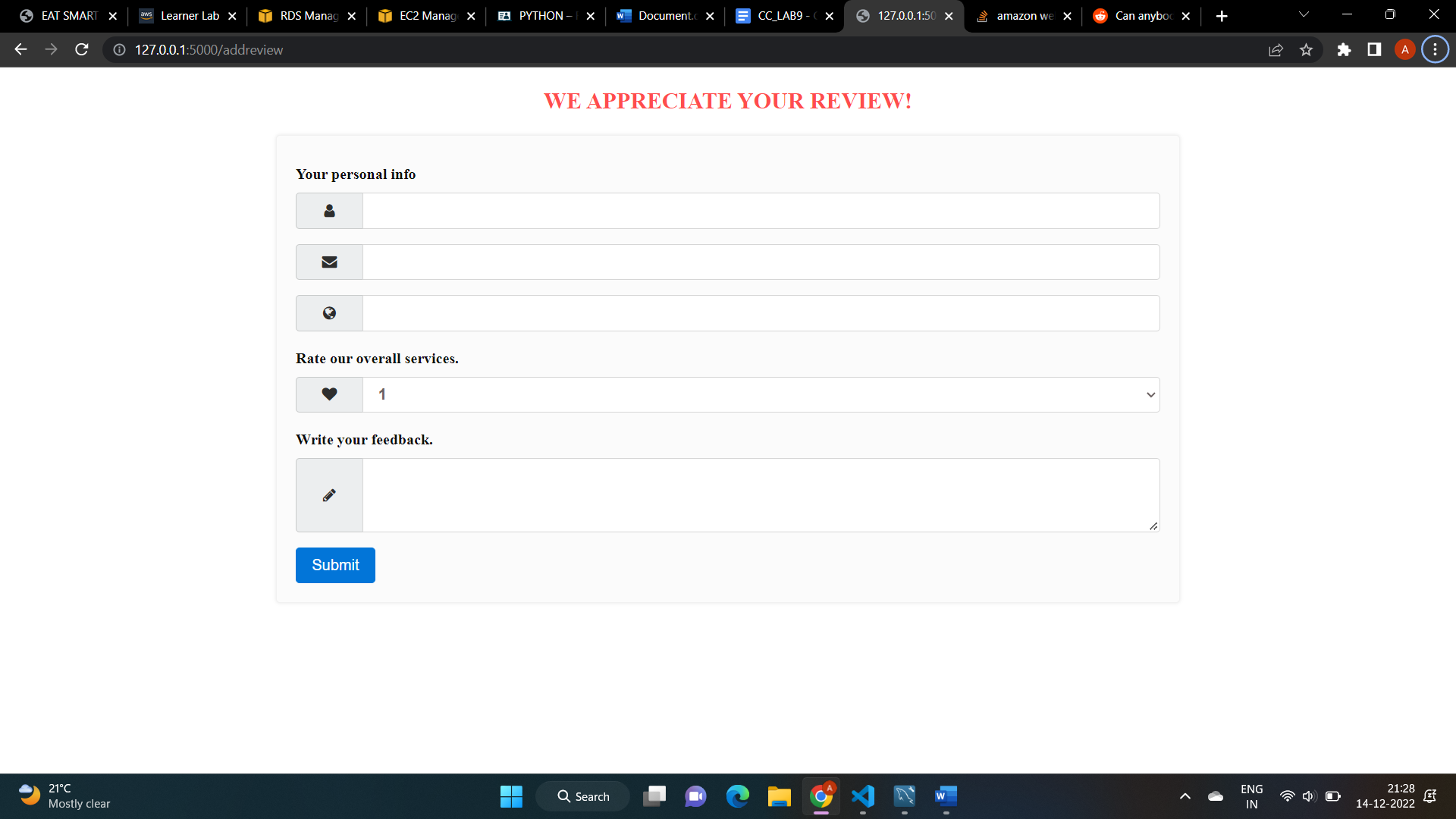
****

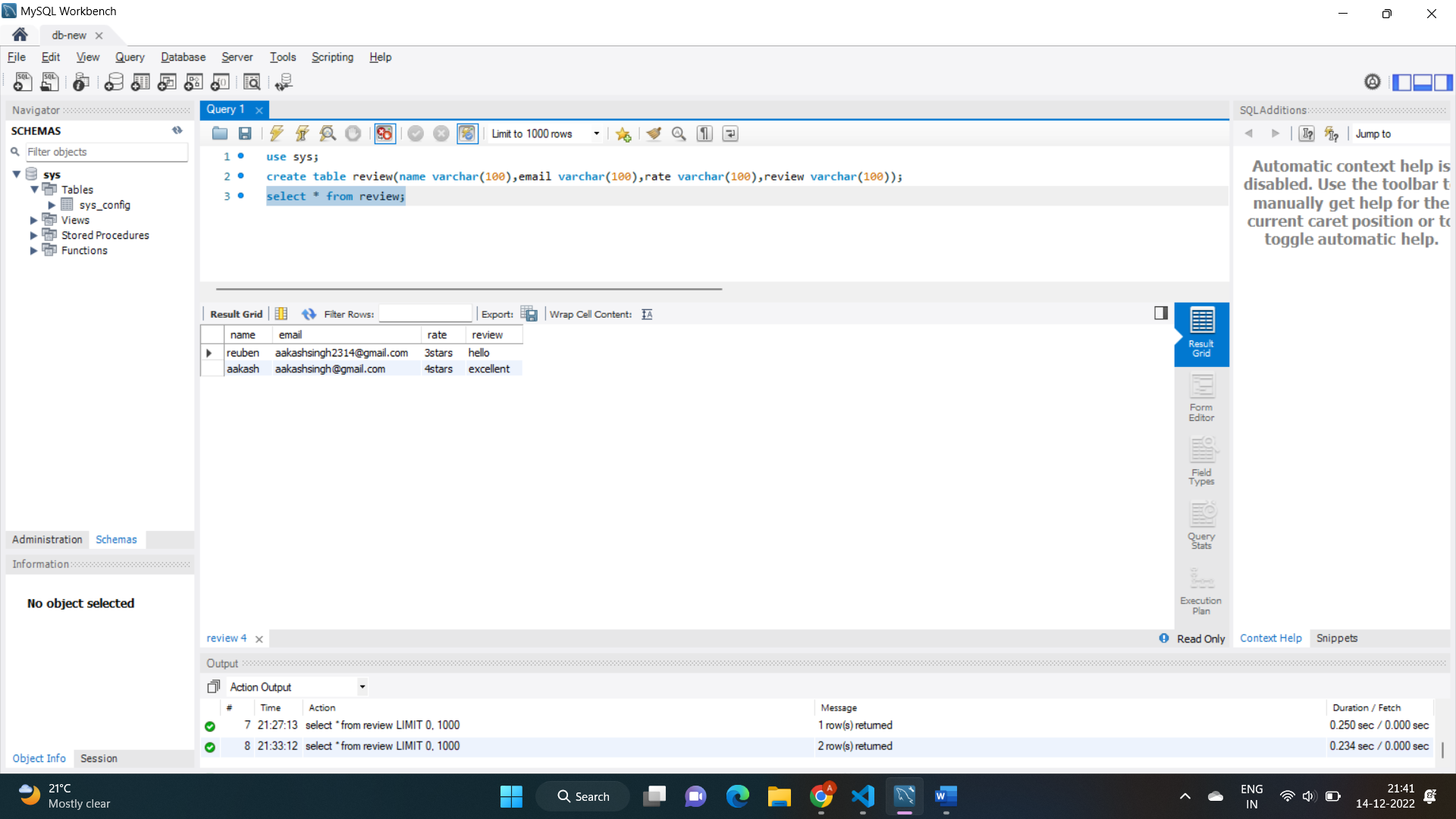
##### 5.4 Login:





**5.5 Submit a Review**

****

****

**6. CONCLUSION AND LEARNINGS**

###### 6.1 Conclusion

The RRS main functions include:

(1) Search: You can search restaurants by restaurant name, type, area, price, rating, recommendation, and distance. Searching in many ways will be easier for you to find the exact restaurant you are interested in.

(2) Rate: This function helps you to see the restaurant's grade clearly from people. You can check it by food ratings and atmosphere ratings. Also you can give ratings for your opinions to people.

(3) Recommendation: This function is like a bulletin. You can leave a message about the restaurant you would like to recommend or a message of the question about the restaurant.

The objective of the RRS project is to let people check great restaurants through the website.’They can get the restaurant information and price here, so that they don't need to call restaurants or check the yellow page for 94 details. They can find great restaurants by people's rating or recommendation to get suggestions. Those are from people's experience and comparison. In the other way, people try new restaurants from others' suggestions, and the restaurant gets more customers. It is an encouragement for the great restaurants.

**6.2 Learnings**

The project really helped us learn a lot. Ranging from how to gathering requirements and planning a project to coding and integrating the designed modules. We face challenges at every level of development. The major one was to keep everyone on the same page while working out the idea.

In terms of the execution of the project, it was a wonderful experience working with Machine Learning, HTML, MySQL and Django to design the server side of the application. Overall, it was a great learning experience.