SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU- 3

(An Autonomous institution affiliated to Visvesvaraya Technological University- Belagavi, Approved by AICTE, Accredited by NAAC with 'A++' Grade, Awarded Diamond College Rating by QS I-GAUGE & ISO 9001:2015 certified)

MINI PROJECT REPORT

ON

"Machine Learning Based User Service Recommendation"

submitted in the partial fulfilment of the requirements for V semester,
Bachelor of Engineering in Artificial Intelligence and Data Science

By

SAHIL RAJ	**1SI20AD022**
P JYOTHIKA CHOWDARY	**1SI20AD014**
SACHITHA D	**1SI20AD021**
SHUBHAM KUMAR GUPTA	**1SI20AD026**

Batch Id: AD10

Under the guidance of

Dr. Srinivasa K Ph.D.
Assistant Professor

Department of Computer Science and Engineering

(Program Accredited by NBA)

Branch: Artificial Intelligence and Data Science

Academic Year: 2022-23

Siddaganga Institute of Technology, Tumkur-3

(An Autonomous institution affiliated to Visvesvaraya Technological University- Belagavi, Approved by AICTE, Accredited by NAAC with 'A++' Grade, Awarded Diamond College Rating by QS I-GAUGE & ISO 9001:2015 certified)

Department of Computer Science and Engineering

(Program Accredited by NBA)

CERTIFICATE

This is to certify that the mini project entitled "Machine Learning Based User Recommendation System" is a bonafide work carried out by **Sahil Raj(1SI20AD022), Jyothika Chowdary Pamulapati(1SI20AD014), Shubham Kumar Gupta(1SI20AD026), Sachitha D(1SI20AD021)** of VI semester **Artificial Intelligence and Data Science**, **SIDDAGANGA INSTITUTE OF TECHNOLOGY** during the academic year 2022-2023.

Signature of the Guide
Dr. Srinivasa K Ph.D.
Assistant Professor

Signature of the Convener
Thejaswini S M.Tech
AssistantProfessor

Signature of the HOD

Dr. A S Poornima
Prof. and Head, Dept. of CSE

Name of the Examiners: **Signature with Date**

1. **Prof.**

2. **Prof.**

ACKNOWLEDGEMENT

We would like to extend our sincere appreciation and respect to his holiness Dr. Sree **Sree Shivakumara Swamigalu**, the esteemed Founder of Sree Siddaganga Education Society, and **Sree Sree Siddalinga Swamigalu**, the President of Sree Siddaganga Education Society, for their blessings.

First and foremost, we would like to express our profound gratitude to **Siddaganga Institute of Technology**, our institution, for providing us with the opportunity to successfully complete our project.

We are thankful to **Dr. M. N. Channabasappa**, the Director, and **Dr. Shivakumaraiah**, the CEO of Siddaganga Institute of Technology-Tumakuru, for their kind cooperation and support.

We would like to convey our appreciation to **Dr. S V Dinesh**, the Principal of Siddaganga Institute of Technology, Tumakuru, for his encouragement towards fostering a positive attitude among students.

We extend our sincere thanks to **Dr. A S Poornima,** Professor and Head of the Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumakuru, for her valuable suggestions and advice.

Our heartfelt gratitude goes to our project guide, **Dr. Srinivasa K**, Assistant Professor in the Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumakuru, for his guidance in successfully completing our mini-project.

We would also like to express our gratitude to **Mrs. Thejaswini S,** Assistant Professor in the Department of Computer Science and Engineering, our mini-project convener, for allowing us to choose this idea for our Mini Project.

We are aware that we have received assistance in various forms from the **faculty, technical staff, and supporting staff** of the Department of Computer Science and Engineering, and we are thankful to all of them for their cooperation.

Lastly, we would like to extend our heartfelt thanks to our **parents and friends** for their constant moral support and encouragement throughout this endeavor.

ABSTRACT

In this work a website is developed to represent a machine learning-powered user service recommendation platform that offers convenient access to a variety of services tailored to individual needs. This website employs machine learning, a subset of artificial intelligence, to generate effective recommendations efficiently. The platform features a user-friendly design, enabling easy access to services such as plumbing, electrical work, carpentry, salon services, therapy, and cleaning. Additionally, users can communicate with service providers directly through the website's messaging feature.

The web-application utilizes Django as a web framework, ensuring a comprehensive project. Furthermore, the integration of the Machine learning framework facilitates rapid development and cross-platform compatibility, while the Google Maps API provides real-time maps and location-based services. This report aims to provide a comprehensive overview of the machine learning-based user service website, outlining its features, benefits, and the roles played by Machine learning and the Google Maps API in its development. The report also explores the website's impact on users. Ultimately, the machine learning-based user service recommendation platform offers businesses a potent tool to enhance customer experience and increase revenue. By harnessing advanced algorithms and data analysis, it delivers personalized and effective recommendations that foster customer satisfaction and loyalty to the brand.

TABLE OF CONTENTS

Contents	Page no
Acknowledgment	i
Abstract	ii
Chapter 1-INTRODUCTION	4
Chapter 2-LITERATURE SURVEY	5-7
Chapter 3-PROPOSED SOLUTION AND EVALUATION	8-9
Chapter 4-PROBLEM STATEMENT AND OBJECTIVES	10
Chapter 5-HIGH LEVEL DESIGN	11
Chapter 6-SYSTEM DESIGN	12
Chapter 7-TOOLS AND TECHNOLOGIES	13-14
Chapter 8-IMPLEMENTATION	15-22
Chapter 10-RESULTS	23
Chapter 9-SNAPSHOTS	24-27
Chapter 11-CONCLUSION	28
References	29-30

CHAPTER 1: INTRODUCTION

The platform of machine learning-based user service recommendation offers personalized service recommendations to users based on their specific needs and preferences. Through this website, customers can conveniently access a wide range of services according to their location and preferred timing. This platform utilizes machine learning, a subset of artificial intelligence, to generate highly efficient and effective recommendations [3].

The user-friendly design of the website facilitates easy access to various services. Users can seek assistance from professionals such as plumbers, electricians, carpenters, salon stylists, therapists, and cleaners. Furthermore, the website includes a messaging feature that enables instant communication between users and service providers.

The web-application employs Django as its web framework, allowing for the creation of a comprehensive project. The machine learning framework provides a rapid and efficient development environment that ensures compatibility across different platforms. Additionally, the integration of the Google Maps API ensures real-time and accurate mapping services based on user location.

This work we developed is a comprehensive overview of the machine learning-based user service website, outlining its features, benefits, and the pivotal role played by machine learning and the Google Maps API in its development. Furthermore, the report will discuss the impact of the website on user experiences.

Overall, the implementation of machine learning-based user service recommendation serves as a potent tool for businesses aiming to enhance customer satisfaction and boost revenue. By leveraging advanced algorithms and data analysis, this platform offers personalized and effective recommendations, thereby improving user satisfaction and fostering brand loyalty

CHAPTER 2: LITERATURE SURVEY

This literature survey provides a comprehensive overview of the recent studies on Recommendation systems based on Machine Learning algorithms and highlights the key findings and contributions of each study. The study provide the overview of Accuracy, Recall, Scalability, Speed and Real-time Updates possible and Interpretability of different types of Recommendation Systems.

➢ Book Recommender System Using Singular Value Decomposition Combined with Slope One Algorithm

Authors: Christina and Z. K. A. Baizal

This study aims to apply the Slope-SVD algorithm to a book recommender system using a rating dataset from the Goodreads web. The performance of the system was tested with the Mean Square Error (MAE). The test results from this study indicate that combining the Slope One algorithm with SVD gives better results than each Slope One or SVD alone [1].

➢ Recommender system using hybrid approach

Authors: S. Sharma, A. Sharma, Y. Sharma and M. Bhatia

In this paper, a new algorithm Composite Search is proposed that combines few filtering algorithms and presents refined results, eliminating drawbacks of other algorithms. The paper presents approach that processes data and provides more filtered results [2].

➢ A Content Based and Collaborative Filtering Recommender System

Authors: V. Thannimalai and L. Zhang

This research proposes a new recommendation system for recommendation generation based on users' ratings and personal profiles. Motivated by existing studies, firstly we propose item-based collaborative filtering to recommend tourist spots based on users' rating [3].

➢ Matrix Factorization Techniques for Recommender Systems

Authors: Y. Koren, R. Bell and C. Volinsky

As the Netflix Prize competition has demonstrated, matrix factorization models are superior to classic nearest neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels [4]

➢ Applicability of Association Rule Mining in Recommendation System for Big Data Analysis

Authors: V. Agarwal, R. K. Gupta and A. Tiwari

This paper briefly describes the recommendation system process flow, its different filtering techniques, and the challenges faced in designing the recommendation system. Moreover, the author also discusses the role of Association Rule Mining (ARM) in creating the recommendation system for resolving the issues of the ecommerce domain. Furthermore, the applicability of ARM in RS in past research is also discussed [5].

➢ Collaborative Filtering Based on Demographic Attribute Vector

Authors: T. Chen and L. He

This paper proposed an NCT/TF(number of common terms / term frequency) collaborate filtering algorithm Based on demographic vector. First, generates user demographic vector base on the user information (age, occupation, gender).then calculate two users similarity base on previous result. and generate new similar by combine it with cosine or PCC similar And then predict item rates by top N similar neighbors. The experiments show that the quality of recommendations improved, while the new user effort is smaller as no initial rating are asked [6]

➢ An Improved Hybrid and Knowledge Based Recommender System for Accurate Prediction of Movies

Authors: V. Agarwal, R. K. Gupta and A. Tiwari

The proposed recommender system model is applied on MovieLens dataset. The comparative analysis was done against content-based recommender system (CBRS), Pearson correlation based collaborative recommender system (PCRS), Frequency-weighted Pearson Correlation (FPC), Weighted Pearson Correlation (WPC) and hybrid recommender systems (HRS). The average RMSE rate achieved by CBRS, PCRS, FPC, WPC, HRS and the proposed hybrid recommender system are 0.3851, 0.3515, 0.3527, 0.3539, 0.3340 and 0.1987 respectively [7].

COMPARATIVE STUDY

Several types of recommendation engine models are available, as shown in Table 1 and each has its own strengths and weaknesses. Here is a list of 20 recommendation systems, along with their performance based on the specified criteria:

TABLE 1: COMPARISON OF RECOMMENDATION ENGINES

Recommendation System	**Accuracy**	**Recall**	**Scalability**	**Speed**	**Real-time Updates**	**Interpretability**
Collaborative Filtering	High	Medium	High	Fast	Yes	Low
Content-Based Filtering	Medium	High	Medium	Fast	No	Medium
Hybrid Filtering	High	Low	High	Slow	Yes	Low
Matrix Factorization	Low	High	Medium	Fast	Yes	Medium
Association Rule Mining	High	High	Low	Slow	No	High
Demographic Filtering	Medium	Medium	High	Fast	Yes	Low
Knowledge-Based Filtering	High	High	Medium	Fast	Yes	High
Context-Aware Recommendation	Low	Low	Low	Slow	No	Low
Item-Based Collaborative Filtering	Medium	High	High	Fast	Yes	Medium
User-Based Collaborative Filtering	High	Medium	Low	Fast	No	Medium
Singular Value Decomposition (SVD)	Medium	High	Medium	Slow	Yes	Low
Deep Learning-based Recommender Systems	High	Low	High	Fast	Yes	Low
Popularity-Based Recommender Systems	Low	High	Medium	Fast	No	Medium
Reinforcement Learning-based Recommender Systems	High	High	Low	Slow	No	High
Hybrid Ensemble Recommender Systems	Medium	Medium	High	Fast	Yes	Low
Item-to-Item Collaborative Filtering	High	High	Medium	Fast	Yes	High
Knowledge Graph-based Recommender Systems	Low	Low	Low	Slow	No	Low
Hybrid Rule-based Recommender Systems	Medium	High	High	Fast	Yes	Medium
Bayesian Personalized Ranking (BPR)	High	Medium	Low	Fast	No	Medium
Genetic Algorithms-based Recommender Systems	Medium	High	Medium	Slow	Yes	Low

Chapter 3

PROPOSED SOLUTION

Considering the status of the project, as it is presently in the beginning stage, it is being decided to use a content-based filtering method and a Collaborative filtering method as shown in Figure 1. Content-based filtering requires a very small amount of data which makes it obvious to select this method.

The content-based filtering is chosen as the proposed solution as it leverages the characteristics and attributes of items to make recommendations. This approach as shown in Figure 1 aligns well with our goal of providing personalized recommendations to users on their individual preferences and interests.

The collaborative filtering [5] is chosen as the proposed solution as it provides a different approach by considering the behavior and preferences of other similar users. This approach as shown in Figure 1 identifies the patterns and relationships among users based on their interactions with items. It captures the wisdom of the crowd and recommends items that are popular among users who share similar preferences.

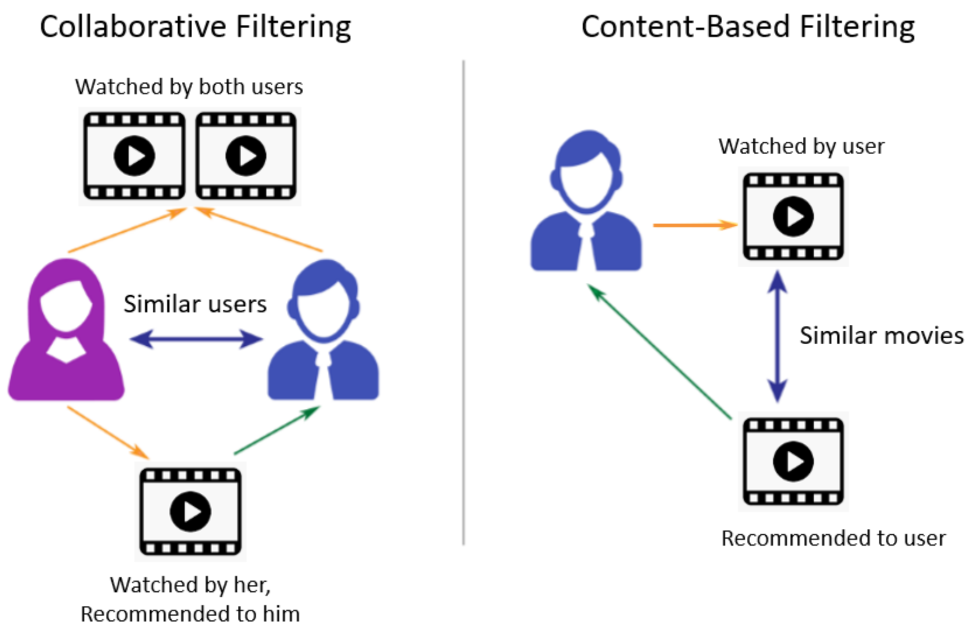


Figure 1 Difference between Content-based filtering and Collaborative filtering

EVALUATION

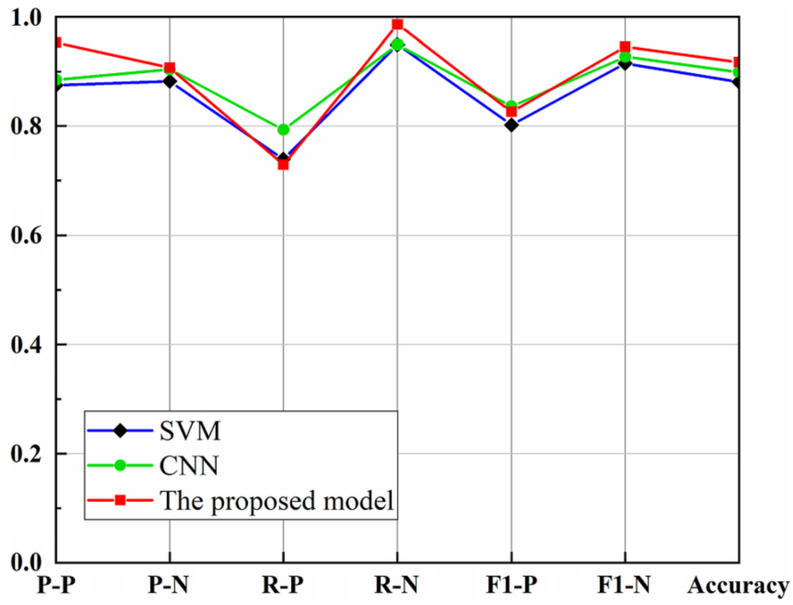


Figure 2 Precision, Recall, F1-score, and Accuracy

In the above shown Figure 2 includes the Precision , Recall ,F1-score and Accuracy of the models SVM, CNN and The proposed model.SVM is a supervised machine learning algorithm commonly used for classification and regression tasks.CNNs are known for their ability to capture local patterns and hierarchical representations.

Precision measures the accuracy of the model's predictions for positive(P-P) and negative(P-N) as shown on Figure 2.P-P represents the precision of how well the model identifies true positive instances and P-N represents the precision of how well the model identifies true negative instances..

Recall measures the model's ability and sensitivity as shown in Figure 2.R-P represents the recall of how well the model captures all the true positive instances and R-N represents the recall of how well the model captures all the true negative instances.

The F1-score is a harmonic mean of precision and recall, providing a balanced measure of the model's performances shown in Figure 2.F1-P represents the F1-score of the positive samples, considering both precision and recall for positive instances.F1-N represents the F1-score of the negative samples, considering both precision and recall for negative instances.

Accuracy measures the overall correctness of the model's predictions across all classes (positive and negative). It calculates the ratio of correctly predicted samples to the total number of samples.

CHAPTER 4

PROBLEM STATEMENT

To create a platform for household needs that helps provide jobs to a greater number of employees while fulfilling the needs of day-to-day activities in every home. The goal is to provide jobs to workers near their native so that they do need not to travel far from their families to sustain their lives. Create a website that can handle this task while implementing a recommendation system for providing better results.

OBJECTIVES

1. **Web Design**: A well-designed user interface (UI) is to be developed, which involves incorporating JavaScript, CSS, and HTML.

2. **Authentication**: To develop an authentication system which requires great security like JWT which will help to keep the password encrypted and the key is always refreshed so decryption of the password becomes nearly impossible.

3. **Databases**: To store the data of each user which includes real-time database storage that is MongoDB which provides a great experience to users.

4. **Recommendation:** To develop a recommendation system that utilizes machine learning techniques that suggest suitable employees to users by considering multiple factors like distance from their current location, price, ratings, and more.

CHAPTER 5: HIGH LEVEL DESIGN

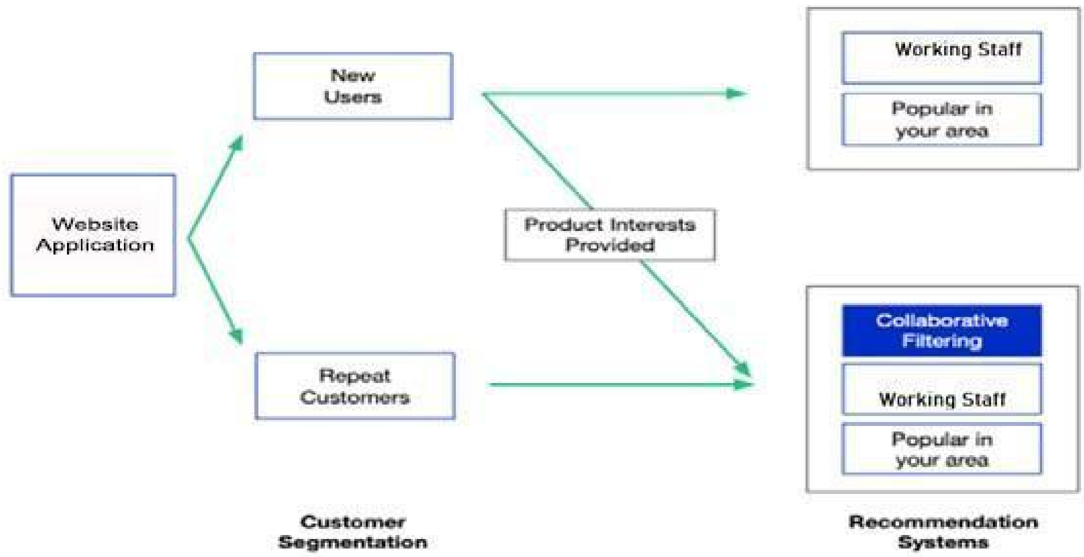


Figure 4 High level diagram

The recommendation system will have two different types of recommendations based on the new users and the past users.

For new users who are unfamiliar with our platform as shown in Figure 4, we employ a content-based recommendation system. This approach analyzes the attributes and characteristics of the services they search for, and suggests similar services based on their preferences. By leveraging the content of the services themselves, we can provide tailored recommendations that align with their initial interests.

Through the collaborative recommendation system, as shown in Figure 4, we leverage the knowledge gained from previous user interactions to offer tailored suggestions to our existing users. By examining their transaction history and identifying users with similar preferences, we can recommend services that have been positively received by others in their peer group. This collaborative approach taps into the collective wisdom of our user community, creating a dynamic and engaging recommendation experience.

CHAPTER 6: SYSTEM DESIGN

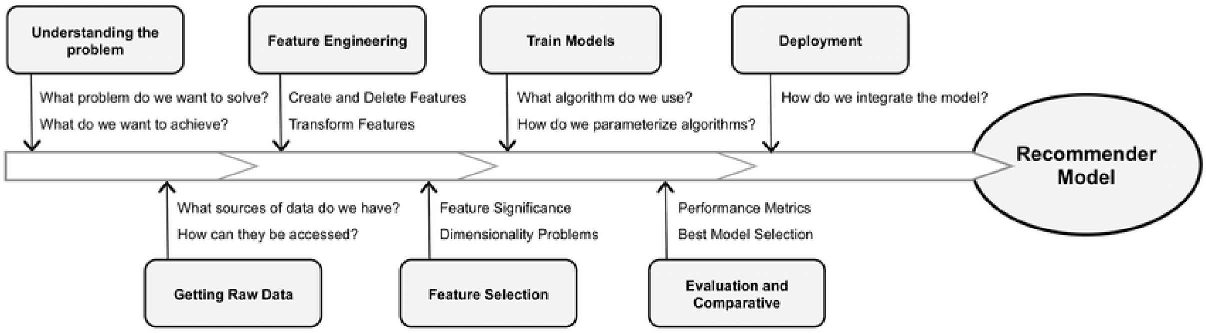


Figure 5 System Design

Understanding the Problem: Define the goals and objectives of the recommender system, such as increasing user engagement or driving sales as shown in Figure 5.

Getting Raw Data: Collect relevant data from various sources, such as user profiles, historical transactions, item attributes, and user feedback.

Feature Engineering: Identify and extract meaningful features from the raw data that can capture user preferences and item characteristics.

Feature Selection: Select the most relevant and informative features that contribute significantly to the recommendation performance.

Train Models: Choose appropriate recommendation algorithms based on the problem type and available data, such as collaborative filtering, matrix factorization, or deep learning models.

Evaluation and Comparative Analysis: Define evaluation metrics, such as precision, recall, or mean average precision, to assess the performance of the recommender system.

Deployment: Integrate the trained model into the production environment, ensuring scalability, efficiency, and real-time response..

CHAPTER 7: TOOLS AND TECHNOLOGIES

1. DJANGO

   For backend- development we have used Django. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

2. JAVASCRIPT

   JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else. (Okay, not everything, but it is amazing what you can achieve with a few lines of JavaScript code.)

3. HTML

   HTML is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.

4. CSS
   CSS is a language of style rules that we use to apply styling to our HTML content, for example setting background colours and fonts, and laying out our content in multiple columns.

5. MONGODB
   MongoDB is a document database used to build highly available and scalable internet applications. With its flexible schema approach, it's popular with development teams using agile methodologies.

6. JWT
   JSON Web Token is an open industry standard used to share information between two entities, usually a client (like your app's frontend) and a server (your app's backend). They contain JSON objects which have the information that needs to be shared.

7. RAZORPAY PAYMENT GATEWAY

   For payments, we will use Razorpay payment integration using API. Payments APIs are used to capture and fetch payments. You can also fetch payments based on orders and card details of payment. You can try out our APIs on the Razorpay Postman Public Workspace.

8. MACHINE LEARNING

   For implementing recommendation for customers. Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. Artificial intelligence systems are used to perform complex tasks in a way that is similar to how humans solve problems. [1]

9. GOOGLE MAPS

   For location services we will use Google Maps API. The Google Maps Platform is a set of APIs and SDKs that allows developers to embed Google Maps into mobile apps and web pages, or to retrieve data from Google Maps.

Chapter 8: IMPLEMENTATION

Recommend_cook (CONTENT-BASED)

NOTE: *Used for recommendation of cook using content based filtering.*

```
FUNCTION recommend_cooks(address, specialty):
    IF address IS None THEN
        PRINT "Address is none"
        RETURN ["None", "None", "None", "None"]
    END IF
    CREATE geolocator OBJECT with user_agent="cook_locator"
    SET user_location TO geolocator.geocode(address + ", India")
    IF user_location IS NOT None THEN
        SET user_lat TO user_location.latitude
        SET user_long TO user_location.longitude
        SET encoded_location TO label_encoder.transform([specialty])[0]
        PRINT df["Specialty"] concatenated with " ||" concatenated with encoded_location
        SET specialty_data TO df WHERE "Specialty" column is equal to encoded_location
        CREATE "Distance" column in specialty_data
        CALCULATE distance BETWEEN [user_lat, user_long] AND specialty_data[["Latitude", "Longitude"]] USING euclidean metric
        MULTIPLY the result by 111
        ASSIGN the result to specialty_data["Distance"]

        SORT specialty_data BY "Distance" in ascending order and "rating" in descending order
        RETURN specialty_data[["Name", "Location", "Distance", "rating"]]
    ELSE
```

```
        PRINT "Geolocation is none"
        RETURN [["None", "None", "None", "None"]]
    END IF
END FUNCTION
FUNCTION main(address, specialty):
    SET recommended_cooks TO CALL recommend_cooks(address, specialty)
    DROP duplicates in recommended_cooks based on "Name" column, keeping the first occurrence
        RETURN the first 10 records of recommended_cooks as a dictionary with "records" orientation
END FUNCTION
CALL main(address, specialty)
```

Recommend_cook (COLLABORATIVE)

NOTE: *Used for recommendation of cook using collaborative filtering.*

```
function load_data():
    ratings = read_csv("ratings.csv")
    cooks = read_csv("cook.csv")
    return ratings, cooks
function data_processing(ratings, cooks):
    n_ratings = length(ratings)
    n_cooks = length(unique(ratings["cookId"]))
    n_users = length(unique(ratings["userId"]))
    mean_rating = ratings.groupby("cookId")[["rating"]].mean()
    return mean_rating
function create_matrix(df):
    N = length(unique(df["userId"]))
    M = length(unique(df["cookId"]))
    user_mapper = create_mapper(unique(df["userId"]), list(range(N)))
    cook_mapper = create_mapper(unique(df["cookId"]), list(range(M)))
    user_inv_mapper = create_mapper(list(range(N)), unique(df["userId"]))
    cook_inv_mapper = create_mapper(list(range(M)), unique(df["cookId"]))
    user_index = create_index_list(df["userId"], user_mapper)
    cook_index = create_index_list(df["cookId"], cook_mapper)
    X = create_csr_matrix(df["rating"], cook_index, user_index, M, N)
    return X, user_mapper, cook_mapper, user_inv_mapper, cook_inv_mapper
function create_mapper(keys, values):
    mapper = create_empty_dictionary()
    for i in range(length(keys)):
```

```
        key = keys[i]
        value = values[i]
        mapper[key] = value
    return mapper
function create_index_list(ids, mapper):
    index_list = create_empty_list()
    for id in ids:
        index = mapper[id]
        add index to index_list
    return index_list
function create_csr_matrix(data, row_index, col_index, M, N):
    X = create_empty_csr_matrix(shape=(M, N))
    for i in range(length(data)):
        rating = data[i]
        row = row_index[i]
        col = col_index[i]
        set_element(X, row, col, rating)
    return X
function find_similar_cooks(cook_id, cook_mapper, cook_inv_mapper, X, k, metric, show_distance):
    neighbour_ids = create_empty_list()
    cook_ind = get_cook_index(cook_id, cook_mapper)
    cook_vec = get_cook_vector(X, cook_ind)
    k = k + 1
    kNN = create_kNN_model(X, metric)
    fit_model(kNN, X)
    cook_vec = reshape_vector(cook_vec)
    neighbour = find_k_nearest_neighbors(kNN, cook_vec, show_distance)
```

```
    for i in range(k):
        n = get_neighbor_item(neighbour, i)
        neighbour_ids.append(get_cook_id(n, cook_inv_mapper))
    remove_first_element(neighbour_ids)
    return neighbour_ids
function get_cook_index(cook_id, cook_mapper):
    return cook_mapper[cook_id]
function get_cook_vector(X, cook_ind):
    return get_element(X, cook_ind)
function create_kNN_model(X, metric):
    kNN = create_empty_kNN_model(metric)
    return kNN
function fit_model(kNN, X):
    fit(kNN, X)
function reshape_vector(cook_vec):
    return reshape(cook_vec, (1, -1))
function find_k_nearest_neighbors(kNN, cook_vec, show_distance):
    return kneighbors(kNN, cook_vec, return_distance=show_distance)
function get_neighbor_item(neighbour, i):
    return get_item(neighbour, i)
function get_cook_id(n, cook_inv_mapper):
    return cook_inv_mapper[n]
function remove_first_element(neighbour_ids):
    remove_first(neighbour_ids)
function main2(cook_id):
    ratings, cooks = load_data()
    mean_rating = data_processing(ratings, cooks)
```

```
X, user_mapper, cook_mapper, user_inv_mapper, cook_inv_mapper = create_matrix(ratings)

cook_Names = create_cook_names_map(cooks)

cook_specialty = create_cook_specialty_map(cooks)

cook_city = create_cook_city_map(cooks)

cook_area = create_cook_area_map(cooks)

cook_pincode = create_cook_pincode_map(cooks)

cook_distict = create_cook_distict_map(cooks)

cook_state = create_cook_state_map(cooks)

similar_ids = find_similar_cooks(cook_id, cook_mapper, cook_inv_mapper, X, k=10)

rval = create_empty_list()

for i in similar_ids:

    val = create_empty_list()

    add_to_list(val, capitalize(cook_Names[i]))

    add_to_list(val, str(round(mean_rating.loc[mean_rating.index == i]["rating"])))

    add_to_list(val, cook_specialty[i])

    add_to_list(val, cook_area[i])

    add_to_list(val, cook_city[i])

    add_to_list(val, cook_distict[i])

    add_to_list(val, cook_state[i])

    add_to_list(val, str(cook_pincode[i]))

    append_to_list(rval, val)

return rval
```

routing

NOTE: *Used for routing the pages in the website*

```
FUNCTION home(request):
    RENDER "home.html" template using request
FUNCTION cook(request):
    RENDER "cook.html" template using request
FUNCTION content_based(request):
    IF request method is "POST" THEN
        SET address to value of "address" field in request POST data
        SET specialty to value of "specialty" field in request POST data
        SET results to the return value of main function with arguments (address, specialty)
        RENDER "content_based.html" template with {"results": results} as context using request
    END IF
    RENDER "content_based.html" template using request
FUNCTION collaborative(request):
    IF request method is "POST" THEN
        SET cook_id to integer value of "cook_id" field in request POST data
        SET results to the return value of main2 function with argument (cook_id)
        RENDER "collaborative.html" template with {"results": results} as context using request
    END IF
    RENDER "collaborative.html" template using request
```

latLongData.py

***NOTE**: For creating Longitude and Latitude of all addresses in database.*

START

READ "cook.csv" into "cook_df" DataFrame

READ "ratings.csv" into "ratings_df" DataFrame

MERGE "cook_df" and "ratings_df" DataFrames on "cookId" column and store the result in "df" DataFrame

CREATE an empty dictionary called "location_dict"

FOR each unique location in the "Location" column of "df":

TRY to do the following:

- Create a geolocator object with user_agent as "cook_locator"

- Use the geolocator to geocode the location

- IF geolocation is not None:

  - Add the location as a key in "location_dict" with its latitude and longitude as the corresponding value

  - PRINT the location, latitude, and longitude

- ELSE:

  - PRINT the location and "None, None"

EXCEPT (in case of any exception):

- Add the location as a key in "location_dict" with None as the value for latitude and longitude

- PRINT the location and "None, None"

MAP the latitude values from "location_dict" to the "Latitude" column of "df"

MAP the longitude values from "location_dict" to the "Longitude" column of "df"

WRITE "df" to a CSV file named "data.csv" without including the index

END

CHAPTER 9: RESULTS

The expected outcome of this project is to develop a robust and efficient machine learning-based user service recommendation platform that effectively meets the needs of both users and service providers. The platform will provide personalized service suggestions to customers based on their specific requirements, ensuring that they receive services that are both effective and efficient.

The project aims to benefit service providers by increasing their revenue and client base. By connecting them with potential customers who are actively seeking their services, the platform will help service providers expand their business and generate more income.

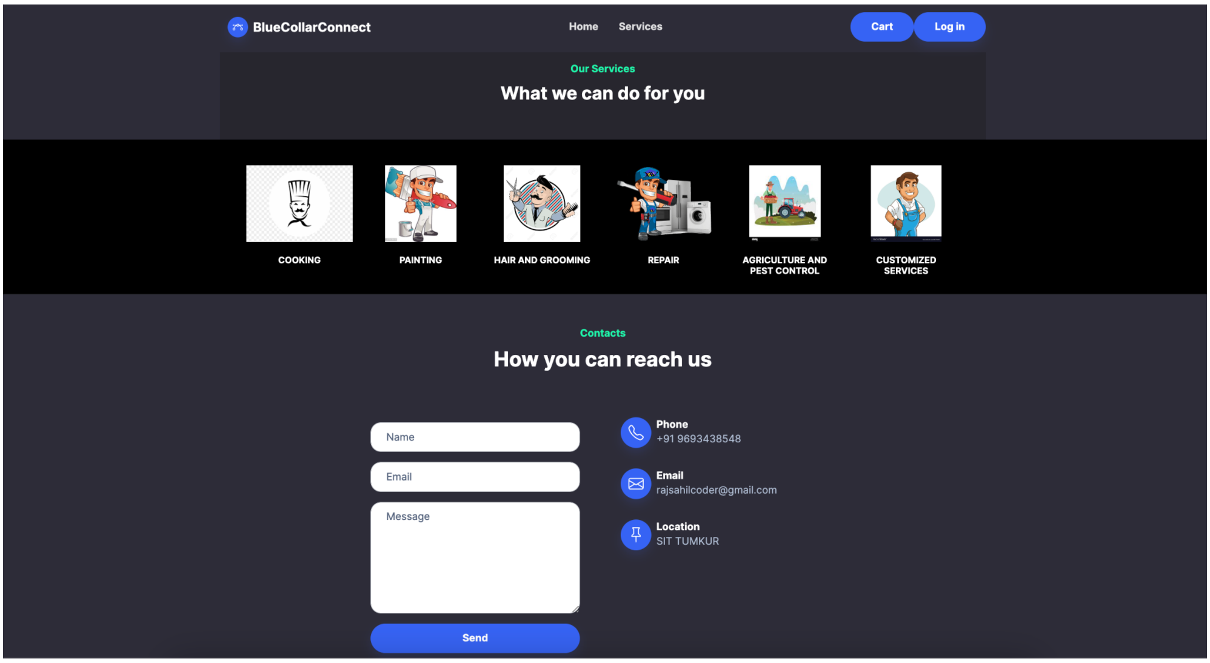
One of the key outcomes of this project is the time and cost-saving aspect for both users and service providers. By recommending relevant services and facilitating instant communication between users and providers, the platform will eliminate the need for users to search extensively for suitable service providers and enable them to directly connect with professionals who can fulfill their requirements. This streamlined process will save time for both parties and reduce unnecessary expenses.

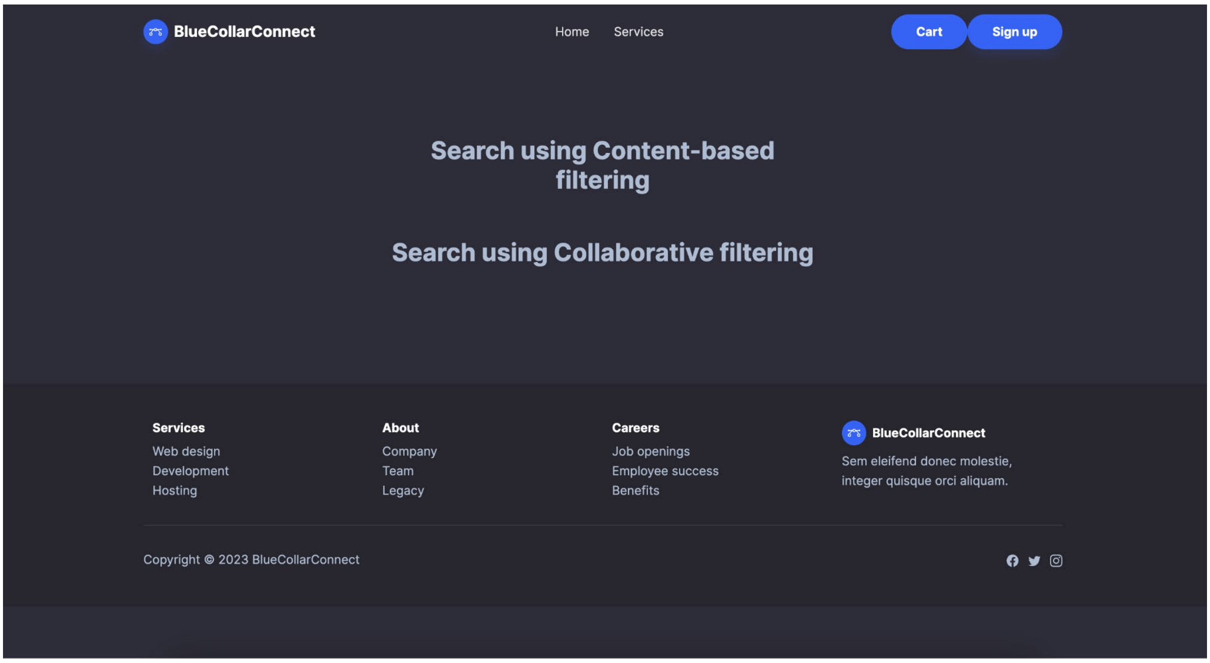
Additionally, the project aims to enhance the user experience by providing recommendations at different stages of the user's decision-making process. By leveraging machine learning algorithms, the platform will analyze user preferences and behavior to offer tailored suggestions that align with their needs and preferences. This personalized approach will improve the overall user experience and increase customer satisfaction.

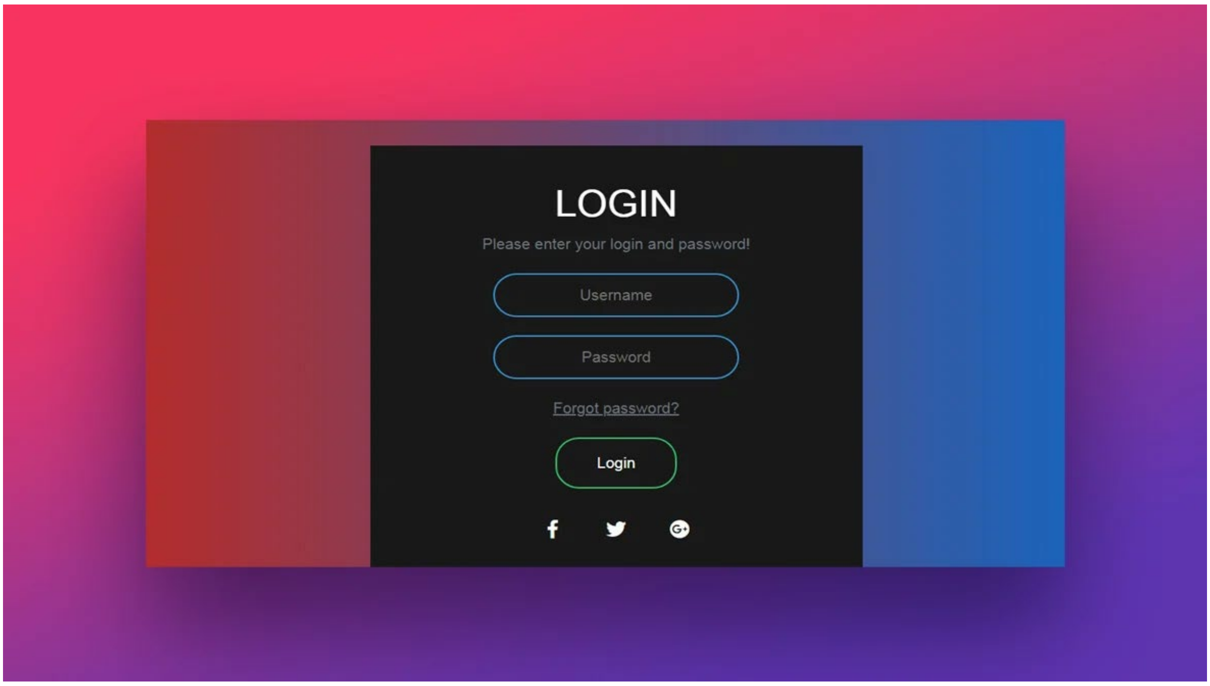
The extensive outcome of this project will be a fully functional web-based application with a user-friendly interface. It will effectively utilize the Django web framework, machine learning algorithms, and the Google Maps API to provide seamless service recommendations and real-time location-based services. The platform will be extensively tested for its performance, accuracy, and reliability to ensure a high-quality user experience.

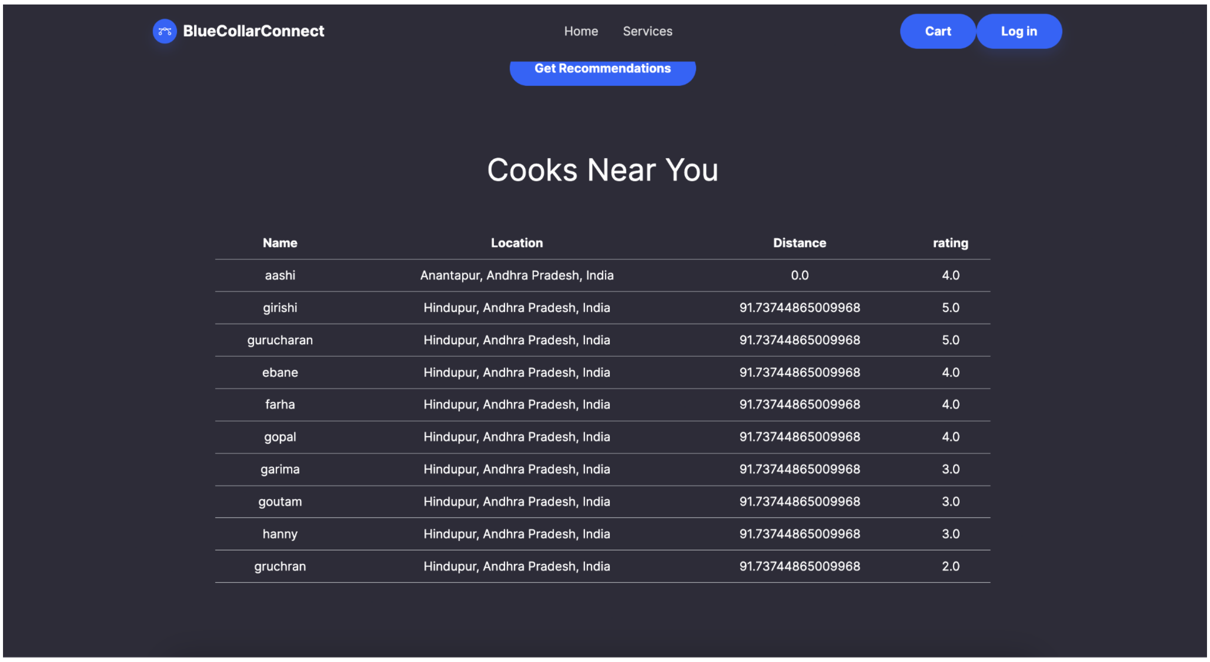
Ultimately, the success of this project will be measured by the satisfaction of both users and service providers. The expected outcome is a platform that not only improves the efficiency and effectiveness of service recommendations but also contributes to the growth and success of businesses in the service industry.

Chapter 10: SNAPSHOTS

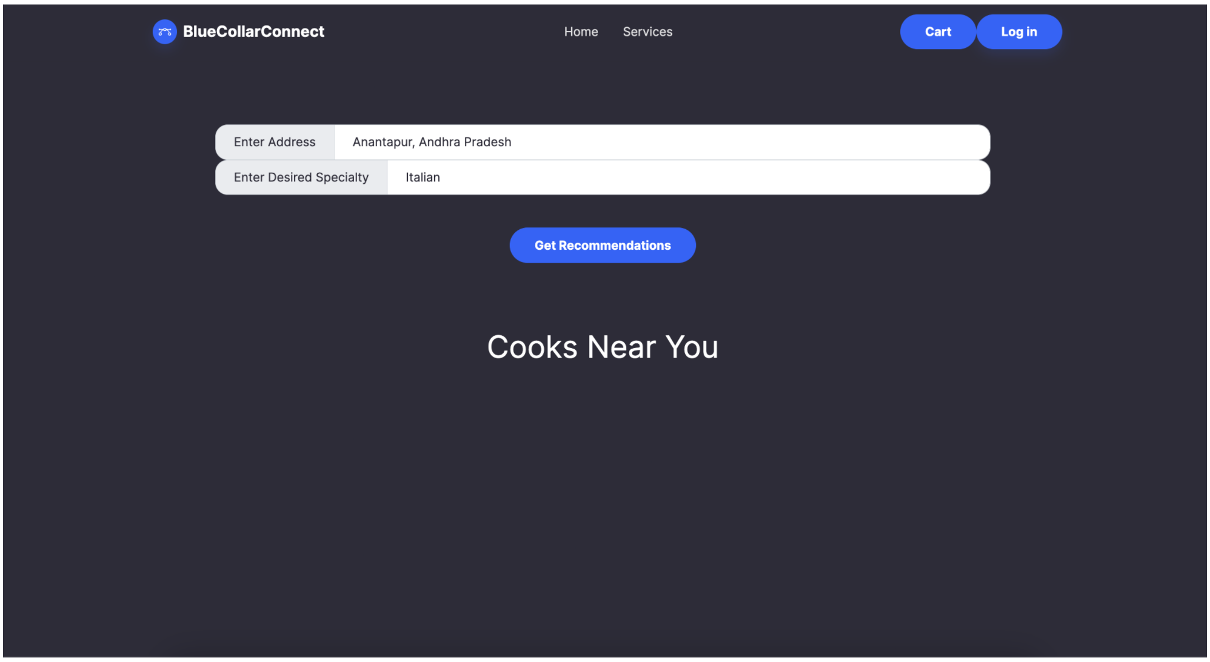


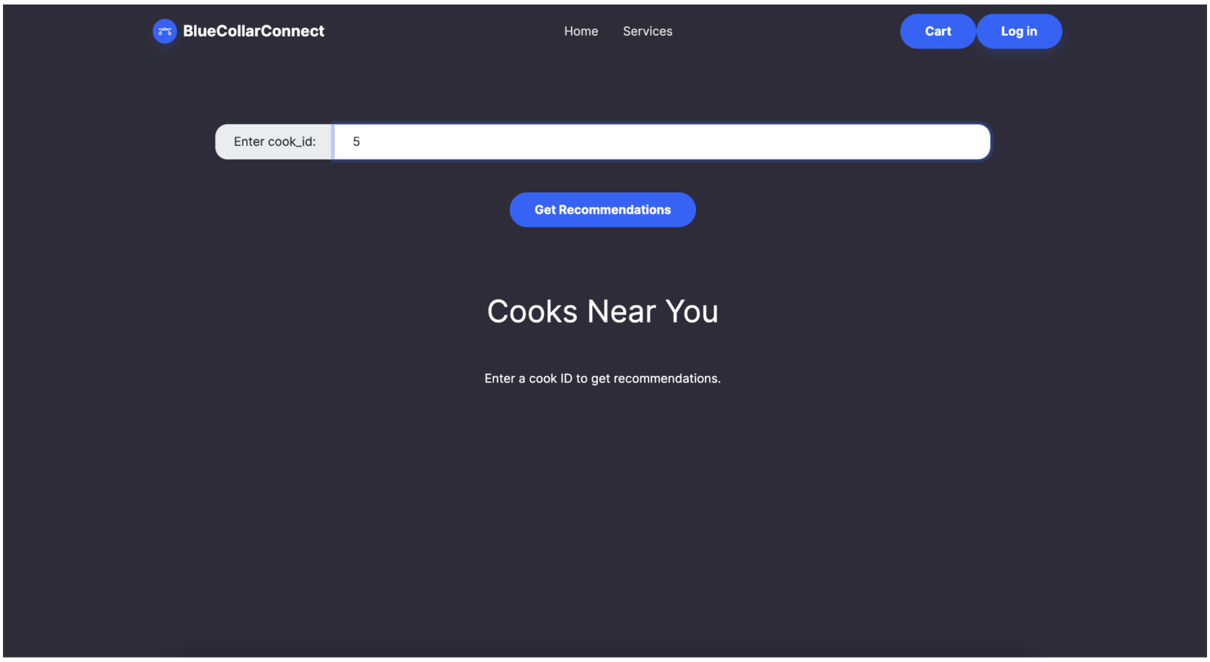


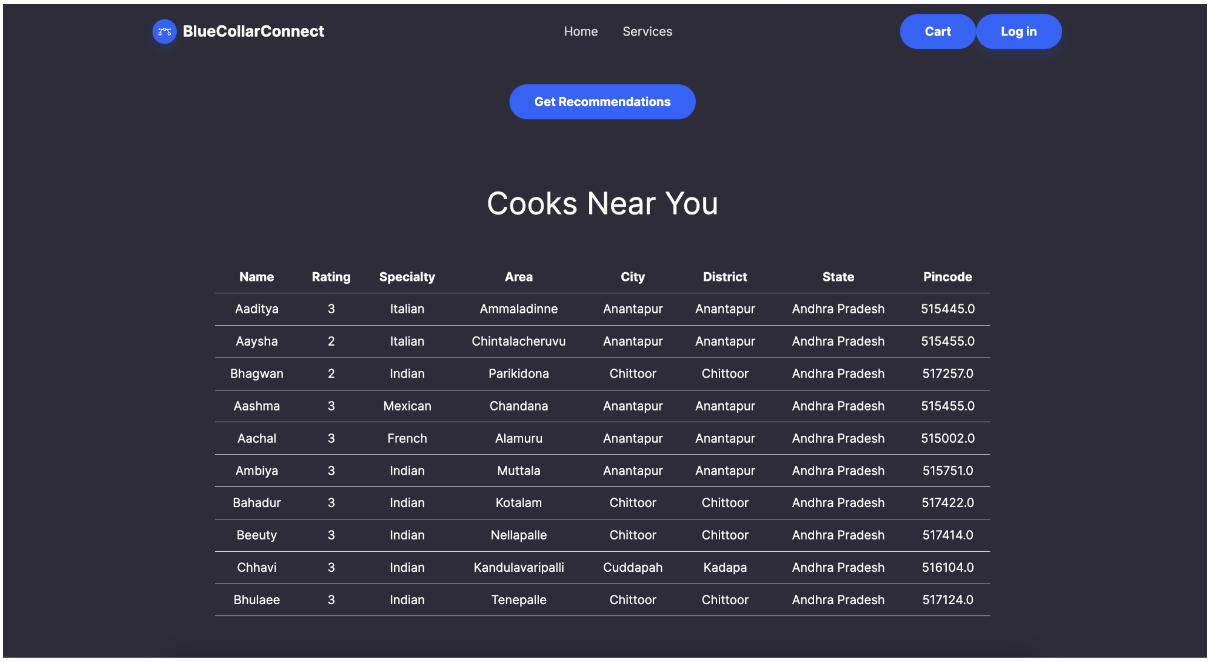




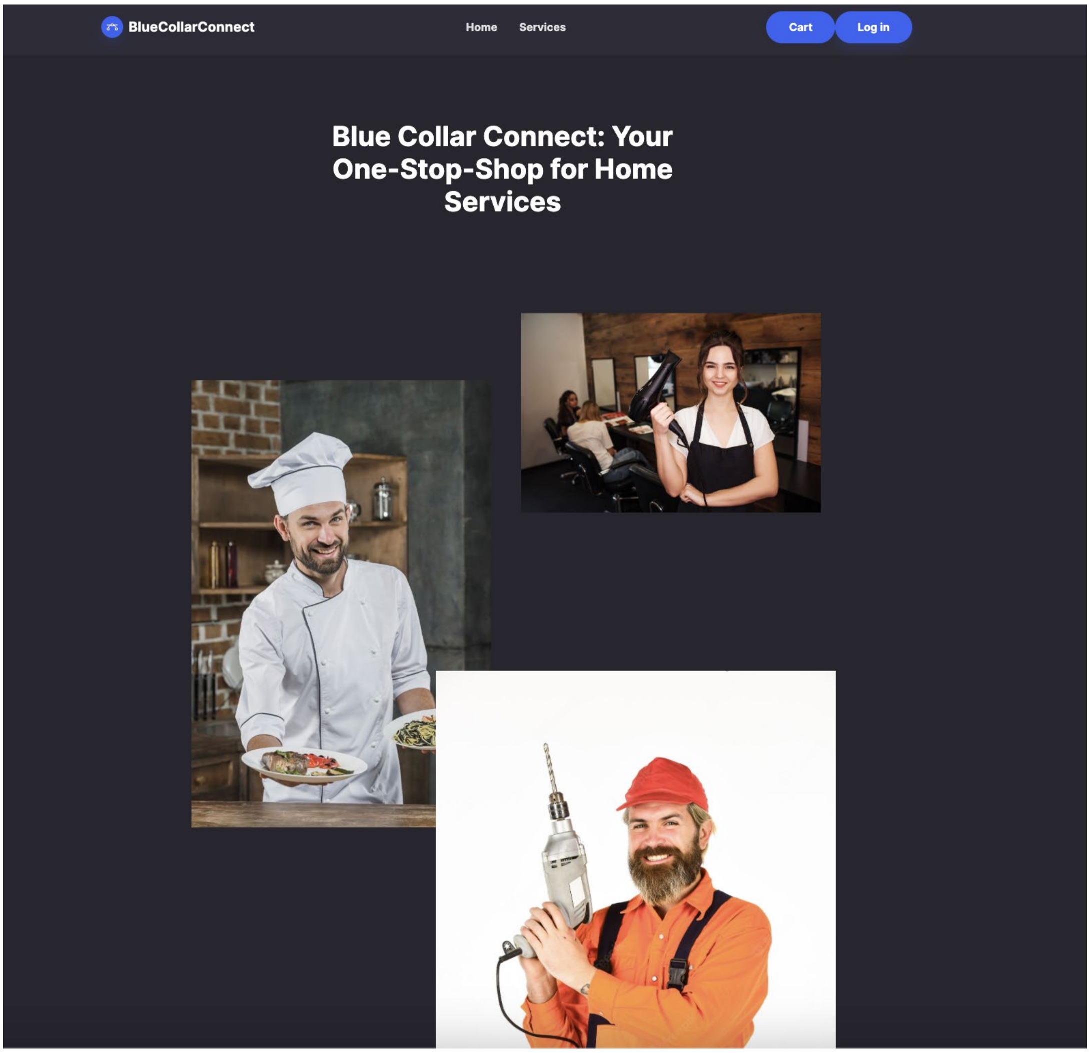
Name	Location	Distance	rating
aashi	Anantapur, Andhra Pradesh, India	0.0	4.0
girishi	Hindupur, Andhra Pradesh, India	91.73744865009968	5.0
gurucharan	Hindupur, Andhra Pradesh, India	91.73744865009968	5.0
ebane	Hindupur, Andhra Pradesh, India	91.73744865009968	4.0
farha	Hindupur, Andhra Pradesh, India	91.73744865009968	4.0
gopal	Hindupur, Andhra Pradesh, India	91.73744865009968	4.0
garima	Hindupur, Andhra Pradesh, India	91.73744865009968	3.0
goutam	Hindupur, Andhra Pradesh, India	91.73744865009968	3.0
hanny	Hindupur, Andhra Pradesh, India	91.73744865009968	3.0
gruchran	Hindupur, Andhra Pradesh, India	91.73744865009968	2.0


BlueCollarConnect
Home
Services
Cart
Log in
Enter Address
Anantapur, Andhra Pradesh
Enter Desired Specialty
Italian
Get Recommendations
Cooks Near You


BlueCollarConnect
Home
Services
Cart
Log in
Enter cook_id:
5
Get Recommendations
Cooks Near You
Enter a cook ID to get recommendations.



Name	Rating	Specialty	Area	City	District	State	Pincode
Aaditya	3	Italian	Ammaladinne	Anantapur	Anantapur	Andhra Pradesh	515445.0
Aaysha	2	Italian	Chintalacheruvu	Anantapur	Anantapur	Andhra Pradesh	515455.0
Bhagwan	2	Indian	Parikidona	Chittoor	Chittoor	Andhra Pradesh	517257.0
Aashma	3	Mexican	Chandana	Anantapur	Anantapur	Andhra Pradesh	515455.0
Aachal	3	French	Alamuru	Anantapur	Anantapur	Andhra Pradesh	515002.0
Ambiya	3	Indian	Muttala	Anantapur	Anantapur	Andhra Pradesh	515751.0
Bahadur	3	Indian	Kotalam	Chittoor	Chittoor	Andhra Pradesh	517422.0
Beeuty	3	Indian	Nellapalle	Chittoor	Chittoor	Andhra Pradesh	517414.0
Chhavi	3	Indian	Kandulavaripalli	Cuddapah	Kadapa	Andhra Pradesh	516104.0
Bhulaee	3	Indian	Tenepalle	Chittoor	Chittoor	Andhra Pradesh	517124.0



CHAPTER 11: CONCLUSION

In conclusion, our project addresses a significant challenge that individuals face when seeking household services, such as cleaning, repairs, and cooking. The difficulty lies in finding reliable workers to complete these tasks efficiently and conveniently. Through the use of cutting-edge technologies and our software expertise, we have developed a solution to alleviate this problem.

Furthermore, our project recognizes the struggles faced by workers from lower and middle-class backgrounds in finding steady employment. These workers often encounter difficulties in securing jobs and may face mistreatment in their workplace. By creating a platform that offers ample job opportunities, we aim to mitigate these issues and positively impact the lives of both service seekers and providers.

By launching this real-life project, we strive to contribute to a society where individuals can easily access reliable services and where workers have greater opportunities for employment. Our motivation lies in alleviating the burdens faced by families and fostering a more inclusive and supportive community.

Through the implementation of cutting-edge technologies and a focus on societal impact, we are confident that our project will make a difference in the lives of many. By bridging the gap between service providers and seekers, we aim to create a more efficient and harmonious ecosystem that benefits everyone involved.

REFERENCES

[1] Christina and Z. K. A. Baizal, "Book Recommender System Using Singular Value Decomposition Combined with Slope One Algorithm," 2022 10th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2022, pp. 346-350, doi: 10.1109/ICoICT55009.2022.9914884.

[2] S. Sharma, A. Sharma, Y. Sharma and M. Bhatia, "Recommender system using hybrid approach," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2016, pp. 219-223, doi: 10.1109/CCAA.2016.7813722.

[3] V. Thannimalai and L. Zhang, "A Content Based and Collaborative Filtering Recommender System," 2021 International Conference on Machine Learning and Cybernetics (ICMLC), Adelaide, Australia, 2021, pp. 1-7, doi: 10.1109/ICMLC54886.2021.9737238.

[4] V. Agarwal, R. K. Gupta and A. Tiwari, "Applicability of Association Rule Mining in Recommendation System for Big Data Analysis," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1810-1813, doi: 10.1109/ICACITE53722.2022.9823695.

[5] T. Chen and L. He, "Collaborative Filtering Based on Demographic Attribute Vector," 2009 ETP International Conference on Future Computer and Communication, Wuhan, China, 2009, pp. 225-229, doi: 10.1109/FCC.2009.68.

[6] D. Khurana and S. Dhingra, "An Improved Hybrid and Knowledge Based Recommender System for Accurate Prediction of Movies," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2021, pp. 881-886.

[7] M. Abusair, M. Sharaf, M. Bozeya and A. Beiruti, "Context-Aware Recommender System based on Content Filtering," 2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2021, pp. 1-6, doi: 10.1109/AICT52784.2021.9620478.

[8] P. Su and H. Ye, "An Item Based Collaborative Filtering Recommendation Algorithm Using Rough Set Prediction," 2009 International Joint Conference on Artificial Intelligence, Hainan, China, 2009, pp. 308-311, doi: 10.1109/JCAI.2009.155.

[9] Z. -D. Zhao and M. -s. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket, Thailand, 2010, pp. 478-481, doi: 10.1109/WKDD.2010.54.

[10] B. Ouhbi, B. Frikh, E. Zemmouri and A. Abbad, "Deep Learning Based Recommender Systems," 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 2018, pp. 161-166, doi: 10.1109/CIST.2018.8596492.

[11] A. Al Jobaer, B. B. Sanchi, A. Javed, M. S. Islam and A. S. M. M. Jameel, "An Advanced Recommendation System by Combining Popularity-Based and User-Based Collaborative Filtering Using Machine Learning," 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 2021, pp. 1-5, doi: 10.1109/ICSCT53883.2021.9642580.

[12] G. Gupta and R. Katarya, "A Study of Deep Reinforcement Learning Based Recommender Systems," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), Jalandhar, India, 2021, pp. 218-220, doi: 10.1109/ICSCCC51823.2021.9478178.

[13] R. N. Laveti, J. Ch, S. N. Pal and N. S. Chandra Babu, "A Hybrid Recommender System Using Weighted Ensemble Similarity Metrics and Digital Filters," 2016 IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW), Hyderabad, India, 2016, pp. 32-38, doi: 10.1109/HiPCW.2016.013.

[14] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in

IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.

[15] R. Zhang, Q. -d. Liu, Chun-Gui, J. -X. Wei and Huiyi-Ma, "Collaborative Filtering for Recommender Systems," 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 2014, pp. 301-308, doi: 10.1109/CBD.2014.47.