A
PROJECT
ON

"Product Recommendation"

Submitted to

Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)
In the Partial Fulfillment of

B.Com. (Computer Application) Final Year

Submitted by
Kalash Jain
Sahil Puri

Under the Guidance of

Pravin J. Yadao

Shiksha Mandal's
G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR
(AUTONOMOUS)
2021-2022

Shiksha Mandal's

G. S. COLLEGE OF COMMERCE & ECONOMICS, NAGPUR (AUTONOMOUS)

CERTIFICATE

(2021 - 2022)

This is to certify that Mr. Kalash Jain and Sahil Puri has completed their project on the topic of Product Recommendation prescribed by G. S. College of Commerce & Economics, Nagpur (Autonomous) for B.Com. (Computer Application) – Semester-VI.

Date: 24/05/2022

Place: Nagpur

Pravin J. Yadao

Project Guide

External Examiner

College of Commerce & Economics NAGPUR

Co-ordinator
Dept. of B.Com. (Comp. Appl.)
G.S.College of Commerce & Economics
Nagpur

Internal Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep gratitude and whole hearted thanks to project guide Prof. Pravin Yadao, Coordinator for his guidance throughout this work. We are very much thankful to him for his constant encouragement, support and kindness.

We are also grateful to our teachers Prof. Rahul Tiwari, Prof. Sushma Gawande, Prof. Preeti Rangari, Prof. Prajkta Deshpande and Prof. Haresh Naringe for their encouragement, help and support from time to time.

We also wish to express our sincere thanks to Principal Dr. N. Y. Khandait for providing us wide range of opportunities, facilities and inspiration to gather professional knowledge and material without which this project could not have been completed.

Date:

Place: Nagpur

Kalash Jain

Sahil Puri

DECLARATION

We **Kalash Jain and Sahil Puri** hereby honestly declare that the work entitled **"Product Recommendation"** submitted by us at G. S. College of Commerce & Economics, Nagpur (Autonomous) in partial fulfillment of requirement for the award of B.Com. (Computer Application) degree by Rashtrasant Tukadoji Maharaj, Nagpur University, Nagpur has not been submitted elsewhere for the award of any degree, during the academic session 2021-2022.

The project has been developed and completed by us independently under the supervision of the subject teacher and project guide.

Kalash Jain

Sahil Puri

Date:

Place: Nagpur

INDEX

1.Introduction

Introduction

Recommender systems assist consumers in discovering information, products and services that are relevant to their needs. Recommender systems are now widely deployed in many settings and many of us routinely consume recommendations from the likes of Amazon, Netflix, and Last.fm, for example.

In order for recommender systems to be effective, the relevance of recommendations made to end-users is key. However, other performance criteria are also important; for example, the ability of the recommender system to make novel and diverse recommendations, and to provide high coverage from a catalogue and user perspective, to name a few.

Recommender systems not only make it easier and more convenient for people to receive information, but also provide great potential in the economic growth as described in. As more and more people realise the importance and power of recommender systems, the exploration for designing high quality recommender systems have been remaining an active topic in the community over the past decade. Due to the continuous efforts into the field, thankfully, many recommender systems have been developed and used in a variety of domains.

Product recommender systems surface items available for purchase across web pages, mobile apps, within emails, or on any connected screens, such as kiosks and various IoT devices. One of the most popular methods used by retailers, recommendations guide visitors to products they are likely interested in, improving the discovery process and helping them find what they want more efficiently.

Today, retailers often have thousands (and sometimes millions) of products in their inventories, making it difficult for consumers to dig up exactly what they are looking for. And with personalized recommendations, brands can help users easily find relevant products based on their affinities, trends, interests, and behavior, with an end goal of driving sales, upsells, cross-sales, larger cart sizes, and higher average order values.

2.Objectives

2.Objectives

Objectives

The aim of this project is to develop a Web application to facilitate a user trying to buy a product (example - a laptop), by recommending them different options before making a purchase decision.

In this implementation, it is important that the system gives choices which are similar to the base requirement - it should give results which fit into a purchase criteria (like budget, etc). While recommending products which are similar, it should also have a wide enough coverage that the user gets multiple options from competing brands.

Overall, the project aims to provide a good platform for users to evaluate and guide them to make a purchase. Based on the project's specification, there are main five objectives extracted. Blow is the brief description of each projective.

1. A web application. A web application is developed to provide GUI interfaces for users to conduct experiments more easily and conveniently. The implementation process goes from prototyping, to the design of pages, to coding, and finally to optimisation. Along the way, the key idea that has been bearing in mind is making the front interfaces as easily interactive as possible.

2. Acquiring a dataset which has information of different types of products with attributes which will be used for judging similarity. In the case of a laptop, these could be screen size, total weight, CPU, graphics card, and price, etc.

3. Implement backend algorithm - This is used by the system to judge similarity and henceforth rank the products based on the user's filled form requirements. This backend will be implemented in Python using various libraries.

4. Identification and application of frontend libraries that make it easy for the Web Application to be displayed correctly on various screens. As a lot of web traffic these days comes from mobile phones, tablets, laptops and desktop computers, it is important to design a system which takes into account proper text sizing and UI layouts when displaying content on different sized screens.

5. Connecting backend algorithm to frontend Web Application by using Python rendering framework Flask. This allows form data to be passed onto Python functions, which can then be processed by the similarity algorithm and pass results back to the frontend Web Application.

3.Preliminary System Analysis

Preliminary System Analysis

Preliminary system analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processor. Gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weakness of the system so as to achieve the organizational goals. System analysis also includes sub-dividing of complex process involving the entire system, identification of data source and annual processes.

1. Preliminary Investigation

In this, process the development team visits the customer and studies their system they investigate the need of the possible software automation in the given system by the end of preliminary investigation, the team furnishes a document that holds a different specific recommendation for the candidate system.it also includes personal assignment cost, project schedule, target dates, main task of the preliminary investigation phase are:

1) Investigation the present system and identify the function to be perform.

2) Identify the objectives of new system in the general, an information system benefits a business increasing efficiency, improving effectiveness, or providing a competitive advantage.

3) Identify problems and suggests a few solutions.

4) It is organized combination of different components.

5) They are independent and inter-related.

2. Present System in Use

Currently many softwares provide the text encryption facility which provides a basic Caesar's Cipher format of algorithm/key to encrypt data.

3. Flaws in present system

Mostly they require a very high subscription amount or the ones that are free aren't that worthy and some do not even allow you to edit info on the go as they only accept files and not text directly.

4. Needs of new system

- **Confidentiality** encodes the message's content without leaking the key
- **Authentication** verifies that the encrypted hasn't been tampered with.
- **Integrity** proves the contents of a message have not been changed since it was encrypted.
- **Nonrepudiation** prevents senders from denying that they encrypted the message as the database is locally stored and it is also inaccessible.

5. Feasibility Study

Feasibility study is the preliminary study undertaken before the real work of the project starts to ascertain the like hood of the project success. It analyzes the possible solutions to a problem and a recommendation on the best solutions to use. It involves the evaluation that how the solution will fit into the corporation. A Feasibility study is defined as an evolution or analysis of the potential impacts of a proposed project or system. A feasibility study is conducted to assist decision makers in determining whether or not to implement a particular project or system.

On the basis of result of the initial study, feasibility study takes place. The feasibility study is basically the proposed system in the lights of its workability,

meeting user's requirements, and effective use of resources and of course, cost effectiveness. The main goal of feasibility study is not to solve the problem but to achieve this scope. In the process of feasibility study, the cost and benefits are estimated with the greater accuracy. It evaluates the benefits of the new system. The feasibility study will contain the extensive data related to financial and operational impact and will include advantage and disadvantages of both current situation and plan.

The aim of feasibility study is to see whether it is possible to develop a reasonable cost. At the end of feasibility study a decision is taken whether or proceed or not.

Feasibility study is to determine various solution of the problem and then picking up one of the best solutions. It is the measure of how beneficial the development of information system will be to an organization. The study also shows the sensitivity of business to change in the basic assumption.

Economic Feasibility

For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluation of the effectiveness of the proposed system.

In this type of feasibility study, the most important is cost and benefit analysis. As the name suggests, it is as analysis of the costs to be incurred in the system and benefits derivable out of the system.

Technical Feasibility

In technical feasibility the following issues are taken into consideration.

- Whether the required technology is available or not.
- Whether the required resources are available like manpower, programmers, testers and debuggers, software and hardware.

Social Feasibility

The affect that a proposed system may have on the social system in the project environment is addressed in the social feasibility. It may happen that particular category of employees may be short or not available as a result of ambient structure. The influence on the social status of the participants by the project should be evaluated on order to guarantee compatibility. It must be identified that the employees in the particular industries may have specific status symbols within the society.

Behavioral Feasibility

It includes how strong the reaction of staff will be towards the development of new system that involves computer's use in their daily work. So resistant to change is identified. It considers human issue. All system development projects introduce change, and people generally resist change. Over resistance from employees may take the form of subrogating the new system (e.g., entering data incorrectly) or deriding the new system to anyone who will listen. Convert resistance typically occurs when employees simply do their jobs using their old methods.

Behavioral feasibility is concerned with assessing the skills and the training needed to use the new is. In some organizations, a proposed system may require mathematical or linguistic skills beyond what the workforce currently processes. In other words, a workforce may simply need to improve their skills. Behavioral feasibility is as much about "can they use it" as it is about "will they use it".

After the feasibility analysis, a "Go/No-Go" decision is reached. The project sponsor and project manager sign off on the decision. If it is a no-go decision, the project is put on the shelf until condition are favorable. Or the project is discarded. Of the decision is "go", then the system development project proceeds.

Project Category

In this project, mainly a GUI module of Python (known as tkinter) has been used for frontend and backend.

It is a windows application.

What is Python? Executive Summary

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Software & Hardware Requirement Specification

Software & Hardware Requirement Specification

1. Hardware -

> Intel/AMD x86 CPU-based system with 2Ghz+ dual cores.

> 4GB of Random Access Memory.

> 1280 x 800 or higher screen resolution.

2. Software -

> At least Windows 10 or MacOS Sierra.

> Python version 3.6+.

> Python Anaconda package library.

> Code editor - Sublime Text or Visual Studio Code.

Detailed System Analysis

Detailed System Analysis

Data Flow Diagram

Depicted below is the Data Flow Diagram (DFD) of the project. It depicts three main pillars of the system design:

1. Preprocessing - This is the step in which the Product dataset (e.g. Laptop dataset) is imported into Python using the Pandas library. This allows us to manipulate the data by removing duplicate rows etc.

2. Algorithm backend - In this step, the processed data flows from a Pandas dataframe into a scikit-learn algorithm (kNN algorithm), which outputs similar products. This needs to be connected to frontend in order to display the results to the user.

3. The frontend or Web Application of the project is made using Bootstrap framework, which runs through Jinja. These framework implementations allow us to insert CSS style into a screen resolution independent end-user display. Flask is used to connect the backend and frontend through Python, as depicted below.

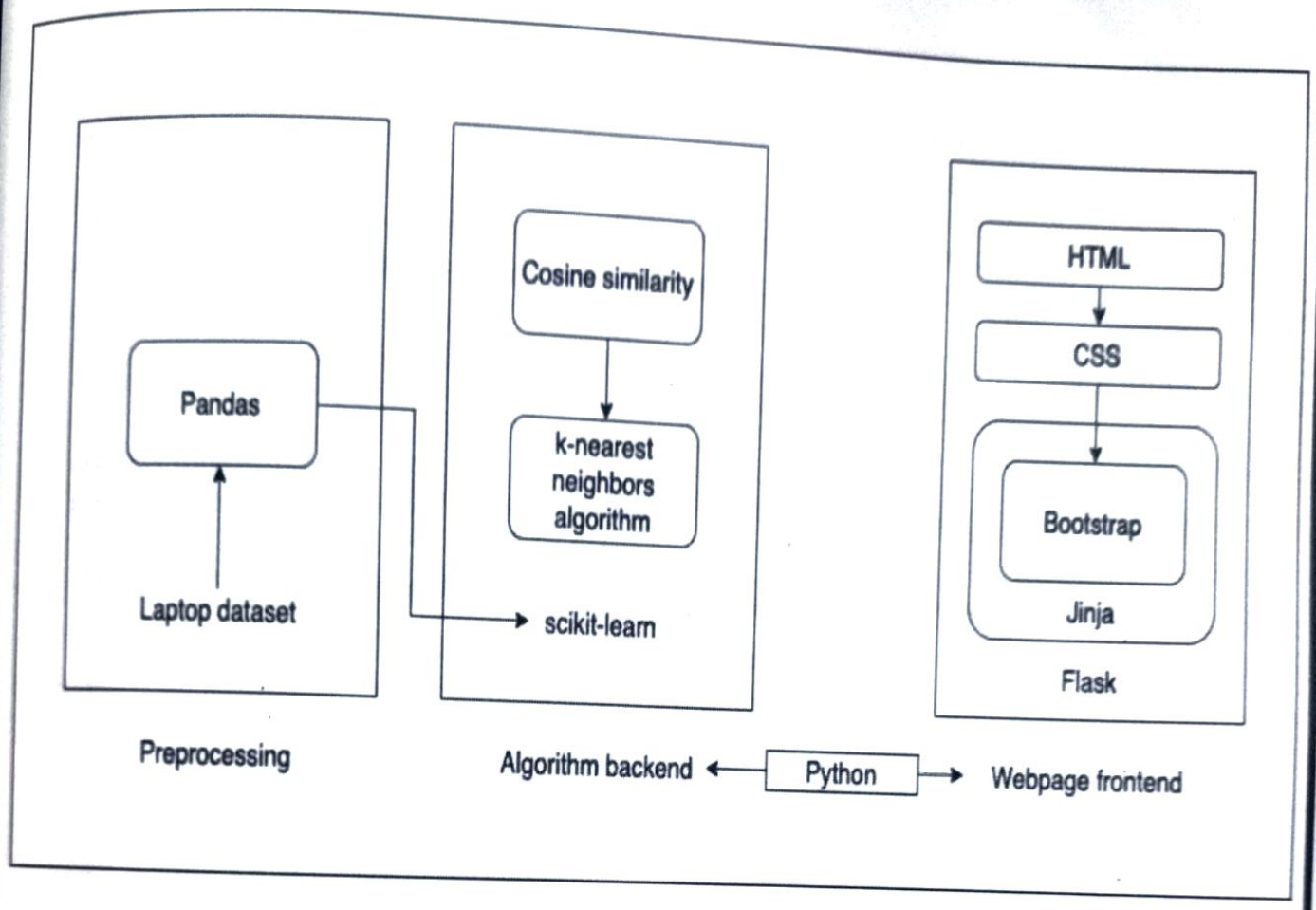


Fig. - Data Flow Diagram

Entity – Relationship Diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database.

ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

An ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

The purpose of ER Diagram is to represent the entity framework infrastructure.

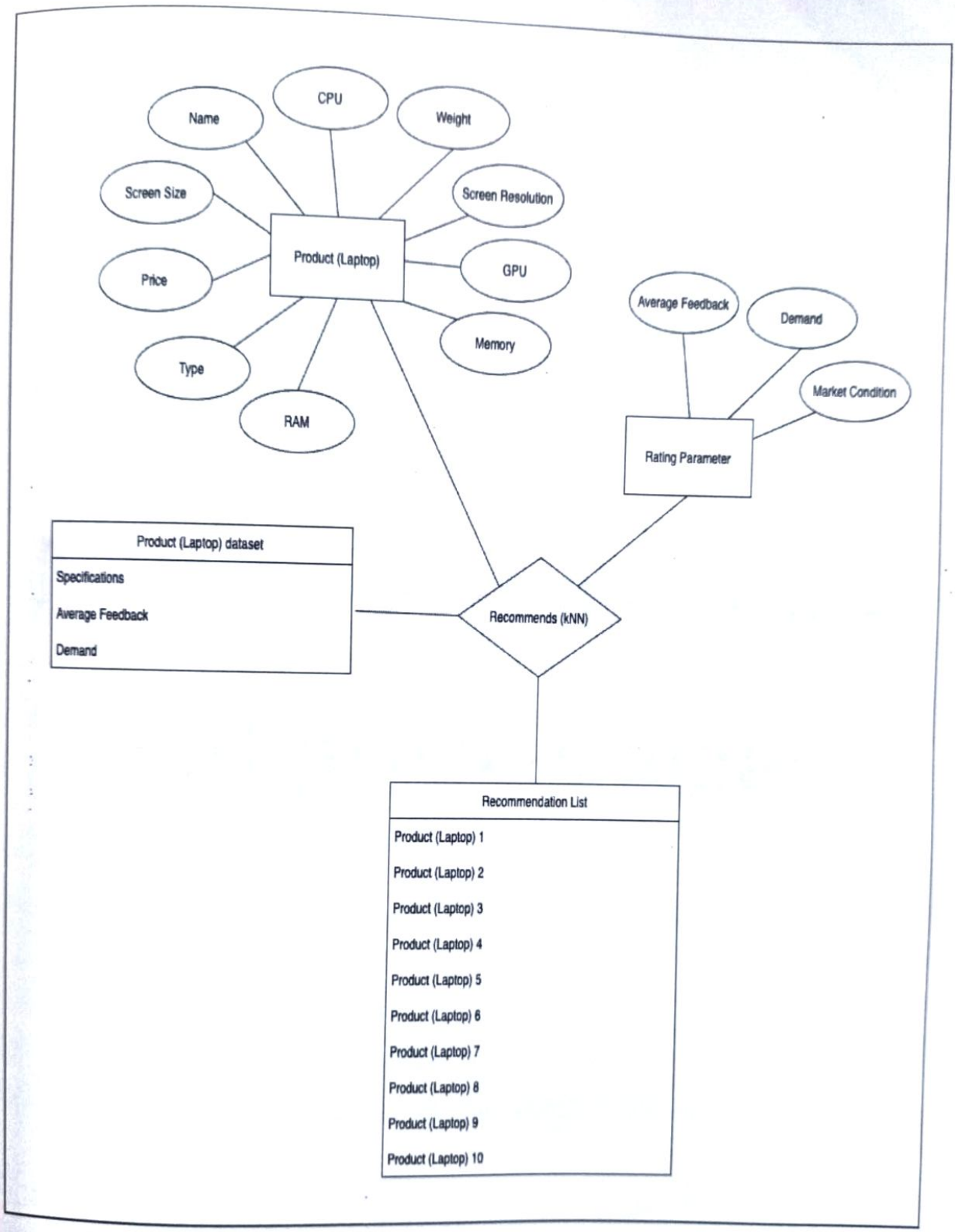


Fig. – Entity Relationship Diagram

Numbers of modules and process logic

Numbers of modules and process logic

5.1 Preprocessing

A preliminary processing of data in order to prepare it for the primary processing or for further analysis. The term can be applied to any first or preparatory processing stage when there are several steps required to prepare data for the user. For example, extracting data from a larger set, filtering it for various reasons and combining sets of data could be preprocessing steps.

Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and Quality of data is first and foremost before running any analysis. Often, data preprocessing is the most important phase of a machine learning project, especially in computational biology. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Examples of data preprocessing include cleaning instance selection, normalization one hotencoding, transformation, feature extraction and selection, etc. The product of data preprocessing is the **No index entries found.**Data preprocessing may affect the way in which outcomes of the final data processing can be interpreted.

5.1.1 Dataset library (pandas)

pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

5.2 Algorithm Backend

Web development activities that are done at the back end of programs are referred to as back end development. Back-end development covers server-side web application logic and integration and activities, like writing APIs, creating libraries, and working with system components instead of frontend development, which focuses on customer-facing services and programs. Backend developers build code that allows a database and an application to communicate with one another. Backend developers take care and maintain the back-end of a website, Including databases, servers, and apps, and they control what you don't see.

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.[3] It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

5.2.1 K-nearest Neighbors Algorithm

The ***k*-nearest neighbors algorithm** (***k*-NN**) is a non-parametric supervised learning method.

It is used for classification and regression. In both cases, the input consists of the k closest training examples in a data set.

k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically.

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

5.2.1.1 Cosine similarity

In data analysis, **cosine similarity** is a measure of similarity between two sequences of numbers. For defining it, the sequences are viewed as vectors in an inner product space, and the cosine similarity is defined as the cosine of the angle between them, that is, the dot product of the vectors divided by the product of their lengths. It follows that the cosine similarity does not depend on the magnitudes of the vectors, but only on their angle. The cosine similarity always belongs to the interval [−1,1]. {\displaystyle [-1,1].}

For example, two proportional vectors have a cosine similarity of 1, two orthogonal vectors have a similarity of 0, and two opposite vectors have a similarity of -1. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1] {\displaystyle [0,1]}.

For example, in information retrieval and text mining, each word is assigned a different coordinate and a document is represented by the vector of the numbers of occurrences of each word in the document. Cosine similarity then gives a useful measure of how similar two documents are likely to be, in terms of their subject matter, and indepently of the length of the documents. The technique is also used to measure cohesion within clusters in the field of data mining.

One advantage of cosine similarity is its low complexity, especially for sparse vectors: only the non-zero coordinates need to be considered.

5.3 Webpage frontend

Webpage frontend

5.3.1 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[2] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.

5.3.1.1 Jinja

Jinja is a web template engine for the Python programming language. It was created by Armin Ronacher and is licensed under a BSD License. Jinja is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated in a sandbox. It is a text-based template language and thus can be used to generate any markup as well as source code.
The Jinja template engine allows customization of tags,[3] filters, tests, and globals.[4] Also, unlike the Django template engine, Jinja allows the template designer to call functions with arguments on objects.

Jinja is a templating language for Python, commonly found in Flask projects. To quote the official documentation, Jinja provides a sandboxed execution environment, a powerful and automated HTML escaping system to prevent XSS attacks, and very usefully, template inheritance.
A Jinja document contains expressions and variables, all of which will get replaced when a document is *rendered*. Let's setup a quick example to get you up and running.

5.3.1.2 Bootstra

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

5.3.1.3 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

5.3.2 HTML

The **HyperText Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

6. System Design

System Design

6.1 Source Code

App.py

```
from flask import Flask, render_template,url_for,request,redirect
from flask_bootstrap import Bootstrap

app = Flask(__name__)
Bootstrap(app)

import os
import pandas as pd
import re
import sys
from sklearn.cluster import KMeans
from sklearn.neighbors import NearestNeighbors
from scipy import spatial
from sklearn import preprocessing
import matplotlib.pyplot as plt
import numpy as np
import math

@app.route('/')
def home():
  return render_template('index.html')

@app.route('/results',methods = ['POST'])
```

```
def result():
    if request.method == 'POST':
        result = request.form

        pivoted_dataframe = pd.read_csv('dataset.csv', encoding='ISO-8859-1')
        pivoted_dataframe = pivoted_dataframe.drop(['Unnamed: 0'], axis=1)

        dataframe_for_clustering = pivoted_dataframe[['TypeName', 'Inches',
'ScreenResolution', 'Ram', 'Memory', 'Weight', 'Price_euros']]

        for i in range(len(dataframe_for_clustering)):
            dataframe_for_clustering['Weight'][i] = re.sub(r"k\S+", "",
dataframe_for_clustering['Weight'][i])    #Regular expression to remove kg
from Weight

        dataframe_for_clustering.TypeName =
pd.Categorical(dataframe_for_clustering.TypeName)
        dataframe_for_clustering.Inches =
pd.Categorical(dataframe_for_clustering.Inches)
        dataframe_for_clustering.ScreenResolution =
pd.Categorical(dataframe_for_clustering.ScreenResolution)
        dataframe_for_clustering.Ram =
pd.Categorical(dataframe_for_clustering.Ram)
        dataframe_for_clustering.Memory =
pd.Categorical(dataframe_for_clustering.Memory)

        for_km = pd.DataFrame()
        for_km['TypeName'] = dataframe_for_clustering.TypeName.cat.codes
        for_km['Inches'] = dataframe_for_clustering.Inches.cat.codes
```

```
    for_km['ScreenResolution'] =
dataframe_for_clustering.ScreenResolution.cat.codes
    for_km['Ram'] = dataframe_for_clustering.Ram.cat.codes
    for_km['Memory'] = dataframe_for_clustering.Memory.cat.codes
    for_km['Weight'] = dataframe_for_clustering['Weight']
    for_km['Price_euros'] = dataframe_for_clustering['Price_euros']

    for_knn = for_km
    scaler = preprocessing.StandardScaler()
    scaled_df = scaler.fit_transform(for_knn)
    scaled_df = pd.DataFrame(scaled_df)
    scaled_df.index = for_knn.index

    no = 20
    knn = NearestNeighbors(n_neighbors=no, algorithm='brute',
metric='cosine').fit(scaled_df)
    forDivide = result['price']
    converted = int(forDivide) / 80
    newProd = pd.Series([ result['type'], result['inches'], result['res'], result['ram'],
result['mem'], result['weight'], converted ])
    newProd = scaler.transform(newProd.values.reshape(1, -1))
    distances, indices = knn.kneighbors(newProd)
    sumx = 0

    diffindex =[]
    diffprod = []
    diffCompany = []
    diffType = []
    diffInch = []
```

```
    diffScreenRes = []
    diffRam = []
    diffMemory = []
    diffWeight = []
    diffPrice = []

    for i in range(no):
        diffindex.append(indices.flatten()[i])
        diffprod.append(pivoted_dataframe['Product'][indices.flatten()[i]])
        diffCompany.append(pivoted_dataframe['Company'][indices.flatten()[i]])
        diffType.append(pivoted_dataframe['TypeName'][indices.flatten()[i]])
        diffInch.append(pivoted_dataframe['Inches'][indices.flatten()[i]])

    diffScreenRes.append(pivoted_dataframe['ScreenResolution'][indices.flatten()[i
    ]])
        diffRam.append(pivoted_dataframe['Ram'][indices.flatten()[i]])
        diffMemory.append(pivoted_dataframe['Memory'][indices.flatten()[i]])
        diffWeight.append(pivoted_dataframe['Weight'][indices.flatten()[i]])

    diffPrice.append(int((pivoted_dataframe['Price_euros'][indices.flatten()[i]]))*80
    )

    return render_template("output.html", result = result, diffindex = diffindex,
    diffprod = diffprod, diffCompany = diffCompany, diffType = diffType, diffInch
    = diffInch, diffScreenRes = diffScreenRes, diffRam = diffRam, diffMemory =
    diffMemory, diffWeight = diffWeight, diffPrice = diffPrice)

@app.route('/allproducts')
def inventory():
```

```
print("here")

pivoted_dataframe = pd.read_csv('dataset.csv', encoding='ISO-8859-1')
pivoted_dataframe = pivoted_dataframe.drop(['Unnamed: 0'], axis=1)

diffindex = []
diffprod = []
diffCompany = []
diffType = []
diffInch = []
diffScreenRes = []
diffRam = []
diffMemory = []
diffWeight = []
diffPrice = []

no = len(pivoted_dataframe)

for i in range(no):
    diffindex.append(i)
    diffprod.append(pivoted_dataframe['Product'][i])
    diffCompany.append(pivoted_dataframe['Company'][i])
    diffType.append(pivoted_dataframe['TypeName'][i])
    diffInch.append(pivoted_dataframe['Inches'][i])
    diffScreenRes.append(pivoted_dataframe['ScreenResolution'][i])
    diffRam.append(pivoted_dataframe['Ram'][i])
    diffMemory.append(pivoted_dataframe['Memory'][i])
    diffWeight.append(pivoted_dataframe['Weight'][i])
```

```
    diffPrice.append(int((pivoted_dataframe['Price_euros'][i]))*80)

  return render_template('allproducts.html', no = no, diffindex = diffindex,
diffprod = diffprod, diffCompany = diffCompany, diffType = diffType, diffInch
= diffInch, diffScreenRes = diffScreenRes, diffRam = diffRam, diffMemory =
diffMemory, diffWeight = diffWeight, diffPrice = diffPrice)

@app.route('/ecommerce')
def ecommerce():

  return render_template('ecommerce.html')

@app.errorhandler(405)
def page_not_found(error):
  return redirect('http://127.0.0.1:5000/')

if __name__=='__main__':
    app.run(debug=True)
```

Output.html

```
<!DOCTYPE html>
<html lang="en">

<head>

 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
 <meta name="description" content="">
 <meta name="author" content="">

 <title>Product Recommendation</title>
 <link href="https://fonts.googleapis.com/css?family=Raleway"
rel="stylesheet">
 <link rel="stylesheet" type="text/css" href="style.css">
 <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8E
RdknLPMO" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
  <link rel="icon" href="{{ url_for('static', filename='icons/icon_alt.ico') }}"
type="image/x-icon">
```

```
<!-- Bootstrap Core CSS -->
<link href="{{ url_for('static',
filename='vendor/bootstrap/css/bootstrap.min.css')}}" rel="stylesheet">

<!-- MetisMenu CSS -->
<link href="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.css')}}" rel="stylesheet">

<!-- Custom CSS -->
<link href="{{ url_for('static', filename='dist/css/sb-admin-2.css')}}"
rel="stylesheet">

<!-- Morris Charts CSS -->
<link href="{{ url_for('static', filename='vendor/morrisjs/morris.css')}}"
rel="stylesheet">

<!-- Custom Fonts -->
<link href="{{ url_for('static', filename='vendor/font-awesome/css/font-
awesome.min.css')}}" rel="stylesheet" type="text/css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
<![endif]-->
```

```
<style media="screen">
.fixed-button {
background-color: #FF2434;
color: #FFFFFF;
border: none;
outline: none;
position: fixed;
bottom: 0px;
right: 15px;
height: 50px;
width: 50px;
margin-bottom: 20px;
border-radius: 30px;
box-shadow: 1px 3px 5px #888888;
}

@keyframes wobble {
 16.65% {
  -webkit-transform: translateY(8px);
  -ms-transform: translateY(8px);
  transform: translateY(8px);
 }
 33.3% {
  -webkit-transform: translateY(-6px);
  -ms-transform: translateY(-6px);
  transform: translateY(-6px);
 }
 49.95% {
  -webkit-transform: translateY(4px);
```

```
    -ms-transform: translateY(4px);
    transform: translateY(4px);
  }
  66.6% {
    -webkit-transform: translateY(-2px);
    -ms-transform: translateY(-2px);
    transform: translateY(-2px);
  }
  83.25% {
    -webkit-transform: translateY(1px);
    -ms-transform: translateY(1px);
    transform: translateY(1px);
  }
  100% {
    -webkit-transform: translateY(0);
    -ms-transform: translateY(0);
    transform: translateY(0);
  }
}

.wobble {
  -webkit-animation-name: wobble;
  animation-name: wobble;
  -webkit-animation-duration: 1s;
  animation-duration: 1s;
  -webkit-animation-timing-function: ease-in-out;
  animation-timing-function: ease-in-out;
  -webkit-animation-iteration-count: 1;
  animation-iteration-count: 1;
```

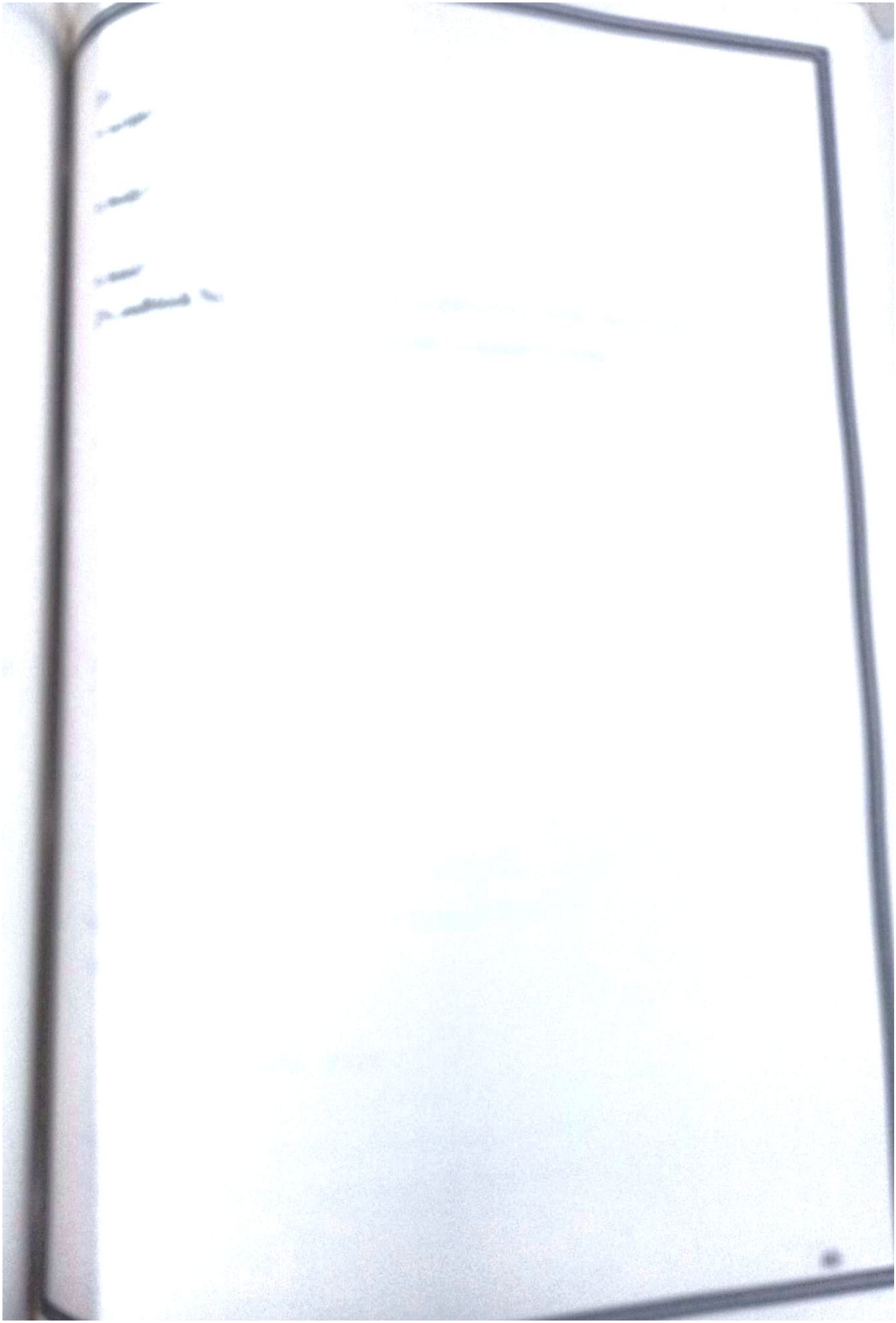
```
<thead>
 <tr>
  <th>Inventory ID</th>
  <th>Company</th>
  <th>Product Name</th>
  <th>Type</th>
  <th>Inches</th>
  <th>Screen Res</th>
  <th>RAM</th>
  <th>Memory</th>
  <th>Weight</th>
  <th>Price (₹)</th>
 </tr>
</thead>
<tbody>
 {% for i in range(20) %}
 <tr>
  <td>{{ diffindex[i] }}</td>
  <td>{{ diffCompany[i] }}</td>
  <td>{{ diffprod[i] }}</td>
  <td>{{ diffType[i] }}</td>
  <td>{{ diffInch[i] }}</td>
  <td>{{ diffScreenRes[i] }}</td>
  <td>{{ diffRam[i] }}</td>
  <td>{{ diffMemory[i] }}</td>
  <td>{{ diffWeight[i] }}</td>
  <td>{{ diffPrice[i] }}</td>
</tr>
{% endfor %}
```

```
        </tbody>
      </table>
    </div>
   </div>
  </div>
 </div>

 <!-- jQuery -->
 <script src="{{ url_for('static', filename='vendor/jquery/jquery.min.js')
}}"></script>
 <!-- Bootstrap Core JavaScript -->
 <script src="{{ url_for('static', filename='vendor/bootstrap/js/bootstrap.min.js')
}}"></script>
 <!-- Metis Menu Plugin JavaScript -->
 <script src="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.js')}}"></script>

 <!-- Morris Charts JavaScript -->
 <script src="{{ url_for('static',
filename='vendor/raphael/raphael.min.js')}}"></script>
 <script src="{{ url_for('static',
filename='vendor/morrisjs/morris.min.js')}}"></script>
 <script src="{{ url_for('static', filename='data/morris-data.js' )}}"></script>

<!-- Custom Theme JavaScript -->
<script src="{{ url_for('static', filename='dist/js/sb-admin-2.js')}}"></script>
<script>
$(document).ready(function(){
 $('[data-toggle="tooltip"]').tooltip();
```



Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta http-equiv="X-UA-Compatible" content="ie=edge">
 <title>Document</title>
</head>
<body>
 <table border = 1>

 {% for key, value in result.items() %}


  <tr>
    <th> {{ key }} </th>
    <td> {{ value }} </td>
   </tr>


 {% endfor %}
</body>
</html>
```

Index.html

```
{% extends "bootstrap/base.html" %}
{% block title %}Product Recommendation{% endblock %}
{% block content %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-
CACHE">
    <link href="https://fonts.googleapis.com/css?family=Raleway"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="style.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<style media="screen">
#myProgress {
    width: 100%;
    background-color: grey;
}
#myBar {
    width: 1%;
    height: 30px;
    background-color: green;
}
</style>
</head>
```

```
<body>
    <div class="jumbotron text-center" style="background-color:
#008CBA; color: #ffffff">
        <div class="row" style="margin-right: 1px;">
            <div class="col-lg-6 ">
                <br>
                    <br>
    <h2>Product Recommendation System</h2>
            </div>
        </div>
</div>
    <div class="container-fluid">
<form action="/results" method="post" style="margin-top: 45px; padding:0
80px 70px 80px;">

        <div class="row">
        <div class="col-lg-6 col-sm-12">
            <div class="form-group" id="name">
            <label for="name">Product Name</label>
            <input type="value" class="form-control" id="name"
name="name" placeholder="Enter Product Name" min="10.0" max="20.0"
required>
            </div>
        </div>
        <div class="col-lg-6">
        <div class="form-group" id="type">
    <label for="type">Product Type</label>
```

```
                <select class="form-control" id="type" name="type"
required>
                                <option value="0">2 in 1
Convertible</option>
                                <option value="1">Gaming</option>
                        <option value="2">Notebook</option>
                        <option value="3">Ultrabook</option>
                    <option value="4">Workstation</option>
                </select>
                    </div>
                        </div>
                    </div>
                    <div class="row">
                    <div class="col-lg-6">
                        <div class="form-group">
                        <label for="newfeat">Important feature for purchase
</label>
                                <select class="form-control" id="newfeat"
name="newfeat"  required >
                                <!-- #1 = Peak(Boom), 2 = Expansion, 3
= Contraction, 4 - Trough(Recession) -->
                                <option onclick="myFunc(this.value)"
value="c1">Product Size</option>
                                <option onclick="myFunc(this.value)"
value="c2">Product Resolution</option>
                                <option onclick="myFunc(this.value)"
value="c3">Product CPU</option>
                                <option onclick="myFunc(this.value)"
value="c4">Product GPU</option>
```

```
                                    <option onclick="myFunc(this.value)"
value="c5">Product Ram</option>
                                    <option onclick="myFunc(this.value)"
value="c6">Product Memory</option>
                                </select>
                            </div>
                            </div>
                            <div class="col-lg-6">
                                <div class="form-group">
                                <label for="newfeed">Importance of
feature</label>
                                <input type="number" class="form-control"
id="newfeed" name="newfeed" data-toggle="tooltip" title="Please enter a value
between 1-10" placeholder="Enter number between 1-10" min="1" max="10"
required>
                                </div>
                            </div>
                        </div>
                            <div class="row">
                        <div class="col-lg-6">
                            <div class="form-group" id="c1">
                                <label for="inches">Screen Size
(Inches)</label>
                                <select class="form-control" id="inches"
name="inches" required>
                                    <option value="0">
    11.6</option>
                                    <option value="1">
    12.5</option>
```

```
                                    <option value="2">
13.3</option>
                                    <option value="3">      14
</option>
                                    <option value="4">      15
</option>
                                    <option value="5">
15.6</option>
                                    <option value="6">
17.3</option>
                                </select>
                        </div>
                </div>
                        <div class="col-lg-6">
                <div class="form-group" id="c2">
                <label  for="res">Screen Resolution</label>
                <select class="form-control" id="res" name="res" required>
                                <option value="0">1366x768</option>
                                <option value="1">4K Ultra HD /
Touchscreen 3840x2160</option>
                                <option value="2">4K Ultra HD
3840x2160</option>
                                <option value="3">Full HD/Touchscreen
1920x1080</option>
                                <option value="4">Full HD
1920x1080</option>
                                <option value="5">IPS Panel
2560x1440</option>
```

```
                                <option value="6">IPS Panel 4K Ultra
HD / Touchscreen 3840x2160</option>
                                <option value="7">IPS Panel 4K Ultra
HD 3840x2160</option>
                                <option value="8">IPS Panel Full
HD Touchscreen 1920x1080</option>
                                <option value="9">IPS Panel Full HD
1920x1080</option>
                                <option value="10">IPS Panel
Touchscreen / 4K Ultra HD 3840x2160</option>
                                <option value="11">Quad HD+ /
Touchscreen 3200x1800</option>
                                <option value="12">Quad HD+
3200x1800</option>
                                <option value="13">Touchscreen / Quad
HD+ 3200x1800</option>
                                <option value="14">Touchscreen
1366x768</option>
                                <option value="15">Touchscreen
2560x1440</option>
                        </select>
                            </div>
                                </div>
                            </div>
                            <div class="row">
                        <div class="col-lg-6">
                            <div class="form-group" id="c3">
                                <label for="cpu">CPU</label>
```

```
<select class="form-control" id="cpu"
name="cpu" required>
                                        <option>  Intel Core i7
7500U 2.7GHz  </option>
                                        <option>  Intel Core i5
7200U 2.5GHz  </option>
                                        <option>  Intel Core i5
8250U 1.6GHz  </option>
                                        <option>  Intel Core i7
7700HQ 2.8GHz  </option>
                                        <option>  Intel Core i3
6006U 2GHz  </option>
                                        <option>  Intel Core i7
8550U 1.8GHz  </option>
                                        <option>  Intel Celeron
Dual Core N3060 1.6GHz  </option>
                                        <option>  Intel Core i5
7440HQ 2.8GHz  </option>
                                        <option>  Intel Core i7
7700HQ 2.7GHz  </option>
                                        <option>  Intel Core i3
7100U 2.4GHz  </option>
                                        <option>  Intel Core i3
6100U 2.3GHz  </option>
                                        <option>  Intel Core i5
7300U 2.6GHz  </option>
                                        <option>  Intel Core i3
6006U 2.0GHz  </option>
```

Dual Core N3060 1.60GHz </option>

8650U 1.9GHz </option>

7600U 2.8GHz </option>

6500U 2.5GHz </option>

6600U 2.6GHz </option>

7560U 2.4GHz </option>

7300HQ 2.5GHz </option>

6700HQ 2.6GHz </option>

7820HK 2.9GHz </option>

7660U 2.5GHz </option>

Quad Core N3710 1.6GHz </option>

7200U 2.7GHz </option>

7Y30 1.0GHz </option>

7Y75 1.3GHz </option>

<option> Intel Celeron

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Core i5

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Core i7

<option> Intel Pentium

<option> Intel Core i5

<option> Intel Core M

<option> Intel Core i7

```
                                                    <option>  Intel Core i5
7Y54 1.2GHz  </option>
                                                    <option>  Intel Core i7
6820HQ 2.7GHz  </option>
                                                    <option>  Intel Xeon E3-
1505M V6 3GHz  </option>
                                                    <option>  Intel Core i7
7820HQ 2.9GHz  </option>
                                                    <option>  Intel Core i5
6440HQ 2.6GHz  </option>
                    </select>
                        </div>
                </div>
                <div class="col-lg-6">
                    <div class="form-group">
                        <label id="c4"for="gpu">GPU</label>
                        <select class="form-control" id="gpu"
name="gpu" required>                                <option>AMD Radeon R7
M445</option>                                       <option>AMD Radeon R5
M430</option>                                       <option>AMD Radeon
520</option>                                        <option>Nvidia GeForce
GTX 1050</option>                                   <option>Intel HD Graphics
520</option>
```

```
530</option>                                    <option>AMD Radeon
620</option>                                    <option>Intel HD Graphics
940MX</option>                                  <option>Nvidia GeForce
Graphics</option>                               <option>Intel HD
930MX</option>                                  <option>Nvidia GeForce
GTX 1060</option>                               <option>Nvidia GeForce
150MX</option>                                  <option>Intel UHD
Graphics 620</option>                           <option>Intel HD Graphics
400</option>                                    <option>Intel Iris Plus
Graphics 640</option>                           <option>Nvidia GeForce
GTX 1070</option>                               <option>Nvidia GeForce
GTX 1050 Ti</option>                            <option>Nvidia GeForce
GTX 1070M</option>                              <option>Nvidia GeForce
```

```
                                        <option>Nvidia GeForce
GTX 960M</option>
                                        <option>Intel HD Graphics
405</option>
                                        <option>Intel HD Graphics
615</option>
                                        <option>Nvidia Quadro
M1200</option>
                                        <option>Nvidia Quadro
M620</option>
                                        <option>AMD FirePro
W5130M</option>
                    </select>
                        </div>
                </div>
            </div>

                <div class="row">
                <div class="col-lg-6">
                    <div class="form-group">
                        <label for="ram" id="c5">RAM</label>
                        <select class="form-control" id="ram"
name="ram" required>
                                        <option value="3">4 GB</option>
                                        <option value="4">6 GB</option>
                                        <option value="5">8
GB</option>
                                        <option value="0">12 GB</option>
                                        <option value="1">16 GB</option>
```

```
                    <option value="2">32 GB</option>
                </select>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="form-group">
                    <label for="mem" id="c5">Memory</label>
                    <select class="form-control" id="mem"
name="mem" required>
                                        <option value="0">1.0TB
Hybrid</option>
                                        <option value="1">128GB
SSD</option>
                                        <option value="2">128GB
SSD + 1TB HDD</option>
                                        <option value="3">128GB
SSD + 2TB HDD</option>
                                        <option value="4">1TB
HDD</option>
                                        <option value="5">1TB
SSD</option>
                                        <option value="6">1TB
SSD + 1TB HDD</option>
                                        <option value="7">256GB
SSD</option>
                                        <option value="8">256GB
SSD + 1TB HDD</option>
                                        <option value="9">256GB
```

```
SSD + 2TB HDD</option>                    <option value="10">256GB

HDD</option>                              <option value="11">2TB

HDD</option>                              <option value="12">500GB

SSD</option>                              <option value="13">512GB

SSD + 1TB HDD</option>                    <option value="14">512GB

SSD + 256GB SSD</option>                  <option value="15">512GB

Flash Storage + 1TB HDD</option>          <option value="16">64GB
                    </select>
                        </div>
                </div>
            </div>

            <div class="row">
                <div class="col-lg-6">
                    <div class="form-group" id="weight">
                        <label for="weight">Weight (kg)</label>
                        <input type="number" class="form-control"
id="weight" name="weight" min="0.00" max="10.00" step="0.1"
placeholder="Enter Weight of Laptop" required>
```

```
                    <div class="form-group">
                        <label for="price">Enter Price (₹)</label>
                        <input type="number" class="form-control"
id="price" name="price" min="0" max="4000000" placeholder="Enter Price of
Laptop (₹)" required>
                    </div>
                </div>
            </div>
                <div class="row">
                 <div class="col-lg-6">
                <div class="form-group">
            <label for="economy">Economy Prices Condition </label>
                <select class="form-control" id="economy"
name="eco" required >
                    <!-- #1 = Peak(Boom), 2 =
Expansion, 3 = Contraction, 4 - Trough(Recession) -->
                    <option value="1">Peak
(Boom)</option>
                    <option
value="2">Expansion</option>
                    <option
value="3">Contraction</option>
                    <option value="4">Trough
(Recession)</option>
                </select>
                </div>
                </div>
                                    "col-lg-6">
```

```
                                    <label for="month">Festive/High
Demand Season? (Yes/No)</label>
                                    <select class="form-control" id="month"
name="month" required >
                                        <option value="1">Yes</option>
                                        <option value="0">No</option>
                                    </select>
                                    </div>
                                </div>
                            </div>
                            <br/>
    <button type="submit" class="btn btn-primary center-
block">Submit</button>
  </form>
 </div>

 <div class='text-center'>
     <button class="btn btn-primary btn-lg"
onclick="window.location.href='/allproducts'" type="button">
         All Products Inventory Database
     </button>
</div>

<br>
```

```
    </button>
</div>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.100.2/js/materialize.mi
n.js"></script>

    <script>

    function myFunc(obj) {
  document.getElementById(obj).style.color = "red";
          var x = document.getElementById(obj).innerHTML
          document.getElementById(obj).innerHTML = x + "( Select the
closest possible value )";

}

    </script>

</body>
</html>
{% endblock %}
```

Ecommerce.html

```
{% extends "bootstrap/base.html" %} {% block title %}Product
Recommendation{% endblock %} {% block content %}

<!DOCTYPE html>
<html lang="en">

<head>

 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
 <meta name="description" content="">
 <meta name="author" content="">

 <title>Product Recommendation</title>
 <link href="https://fonts.googleapis.com/css?family=Raleway"
rel="stylesheet">
 <link rel="stylesheet" type="text/css" href="style.css">
 <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8E
RdknLPMO" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
```

```
<link rel="icon" href="{{ url_for('static', filename='icons/icon_alt.ico') }}"
type="image/x-icon">

<!-- Bootstrap Core CSS -->
<link href="{{ url_for('static',
filename='vendor/bootstrap/css/bootstrap.min.css')}}" rel="stylesheet">

<!-- MetisMenu CSS -->
<link href="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.css')}}" rel="stylesheet">

<!-- Custom CSS -->
<link href="{{ url_for('static', filename='dist/css/sb-admin-2.css')}}"
rel="stylesheet">

<!-- Morris Charts CSS -->
<link href="{{ url_for('static', filename='vendor/morrisjs/morris.css')}}"
rel="stylesheet">

<!-- Custom Fonts -->
<link href="{{ url_for('static', filename='vendor/font-awesome/css/font-
awesome.min.css')}}" rel="stylesheet" type="text/css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if
```

```
    <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
<style media="screen">
.fixed-button {
background-color: #FF2434;
color: #FFFFFF;
border: none;
outline: none;
position: fixed;
bottom: 0px;
right: 15px;
height: 50px;
width: 50px;
margin-bottom: 20px;
border-radius: 30px;
box-shadow: 1px 3px 5px #888888;
}

@keyframes wobble {
16.65% {
 -webkit-transform: translateY(8px);
 -ms-transform: translateY(8px);
 transform: translateY(8px);
}
33.3% {
 -webkit-transform: translateY(-6px);
```

```
}
49.95% {
 -webkit-transform: translateY(4px);
 -ms-transform: translateY(4px);
 transform: translateY(4px);
}
66.6% {
 -webkit-transform: translateY(-2px);
 -ms-transform: translateY(-2px);
 transform: translateY(-2px);
}
83.25% {
 -webkit-transform: translateY(1px);
 -ms-transform: translateY(1px);
 transform: translateY(1px);
}
100% {
 -webkit-transform: translateY(0);
 -ms-transform: translateY(0);
 transform: translateY(0);
}
}

.wobble {
 -webkit-animation-name: wobble;
 animation-name: wobble;
 -webkit-animation-duration: 1s;
```

```
animation-timing-function: ease-in-out;
-webkit-animation-iteration-count: 1;
animation-iteration-count: 1;

}
</style>

</head>

<body>

<button class="fixed-button wobble fixed-action-button" data-toggle="tooltip"
title="Click me to predict for a new product"
onclick="window.location.href='http://127.0.0.1:5000/'" type="button">
 <i class="glyphicon glyphicon-plus"></i>
</button>
<div class="container">
 <div class="row">
  <div class="col-lg-12">
   <h1 class="page-header">All Products Inventory Database </h1>
  </div>
 </div>
  <!-- /.col-lg-12 -->

 <!-- /.row -->

 <!-- /.panel -->
 <div class="panel panel-default">
```

```
</div>
<!-- /.panel-heading -->
<div class="panel-body">


<div class="table-responsive">
 <table class="table table-bordered table-hover table-striped">
  <thead>
   <tr>
    <th>Inventory ID</th>
    <th>Company</th>
    <th>Product Name</th>
    <th>Type</th>
    <th>Inches</th>
    <th>Screen Res</th>
    <th>RAM</th>
    <th>Memory</th>
    <th>Weight</th>
    <th>Price (?)</th>
   </tr>
  </thead>
  <tbody>
   {% for i in range(no) %}
   <tr>
    <td>{{ diffindex[i] }}</td>
```

```
        <td>{{ diffScreenRes[i] }}</td>
        <td>{{ diffRam[i] }}</td>
        <td>{{ diffMemory[i] }}</td>
        <td>{{ diffWeight[i] }}</td>
        <td>{{ diffPrice[i] }}</td>
      </tr>
      {% endfor %}
    </tbody>
   </table>
  </div>
 </div>
</div>
</div>

<!-- jQuery -->
<script src="{{ url_for('static', filename='vendor/jquery/jquery.min.js')
}}"></script>
<!-- Bootstrap Core JavaScript -->
<script src="{{ url_for('static', filename='vendor/bootstrap/js/bootstrap.min.js')
}}"></script>
<!-- Metis Menu Plugin JavaScript -->
<script src="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.js')}}"></script>

<!-- Morris Charts JavaScript -->
<script src="{{ url  for('static',
```

```
<script src="{{ url_for('static', filename='data/morris-data.js' )}}"></script>

<!-- Custom Theme JavaScript -->
<script src="{{ url_for('static', filename='dist/js/sb-admin-2.js')}}"></script>
<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>


</body>


</html>
{% endblock %}
```

Allproducts.html

```
{% extends "bootstrap/base.html" %} {% block title %}Product
Recommendation{% endblock %} {% block content %}

<!DOCTYPE html>
<html lang="en">

<head>

<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
<meta name="description" content="">
<meta name="author" content="">

<title>Product Recommendation</title>
<link href="https://fonts.googleapis.com/css?family=Raleway"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href="style.css">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8E
RdknLPMO" crossorigin="anonymous">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet"
```

```
<link rel="icon" href="{{ url_for('static', filename='icons/icon_alt.ico') }}"
type="image/x-icon">

<!-- Bootstrap Core CSS -->
<link href="{{ url_for('static',
filename='vendor/bootstrap/css/bootstrap.min.css')}}" rel="stylesheet">

<!-- MetisMenu CSS -->
<link href="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.css')}}" rel="stylesheet">

<!-- Custom CSS -->
<link href="{{ url_for('static', filename='dist/css/sb-admin-2.css')}}"
rel="stylesheet">

<!-- Morris Charts CSS -->
<link href="{{ url_for('static', filename='vendor/morrisjs/morris.css')}}"
rel="stylesheet">

<!-- Custom Fonts -->
<link href="{{ url_for('static', filename='vendor/font-awesome/css/font-
awesome.min.css')}}" rel="stylesheet" type="text/css">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
                                        ml5shiv.js"></script>
```

```
    <script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
<style media="screen">
.fixed-button {
background-color: #FF2434;
color: #FFFFFF;
border: none;
outline: none;
position: fixed;
bottom: 0px;
right: 15px;
height: 50px;
width: 50px;
margin-bottom: 20px;
border-radius: 30px;
box-shadow: 1px 3px 5px #888888;
}

@keyframes wobble {
 16.65% {
  -webkit-transform: translateY(8px);
  -ms-transform: translateY(8px);
  transform: translateY(8px);
 }
```

```
}
49.95% {
 -webkit-transform: translateY(4px);
 -ms-transform: translateY(4px);
 transform: translateY(4px);

}
66.6% {
 -webkit-transform: translateY(-2px);
 -ms-transform: translateY(-2px);
 transform: translateY(-2px);

}
83.25% {
 -webkit-transform: translateY(1px);
 -ms-transform: translateY(1px);
 transform: translateY(1px);

}
100% {
 -webkit-transform: translateY(0);
 -ms-transform: translateY(0);
 transform: translateY(0);
 }
}


.wobble {
 -webkit-animation-name: wobble;
```

```
animation-timing-function: ease-in-out;
-webkit-animation-iteration-count: 1;
animation-iteration-count: 1;

}
</style>

</head>

<body>

 <button class="fixed-button wobble fixed-action-button" data-toggle="tooltip"
title="Click me to predict for a new product"
onclick="window.location.href='http://127.0.0.1:5000/'" type="button">
  <i class="glyphicon glyphicon-plus"></i>
 </button>
 <div class="container">
  <div class="row">
   <div class="col-lg-12">
    <h1 class="page-header">All Products Inventory Database </h1>
   </div>
  </div>
   <!-- /.col-lg-12 -->
```

```
</div>
<!-- /.panel-heading -->
<div class="panel-body">


<div class="table-responsive">
 <table class="table table-bordered table-hover table-striped">
  <thead>
   <tr>
    <th>Inventory ID</th>
    <th>Company</th>
    <th>Product Name</th>
    <th>Type</th>
    <th>Inches</th>
    <th>Screen Res</th>
    <th>RAM</th>
    <th>Memory</th>
    <th>Weight</th>
    <th>Price (₹)</th>
   </tr>
  </thead>
  <tbody>
   {% for i in range(no) %}
   <tr>
```

```
        <td>{{ diffScreenRes[i] }}</td>
        <td>{{ diffRam[i] }}</td>
        <td>{{ diffMemory[i] }}</td>
        <td>{{ diffWeight[i] }}</td>
        <td>{{ diffPrice[i] }}</td>
      </tr>
      {% endfor %}
    </tbody>
   </table>
  </div>
 </div>
</div>

<!-- jQuery -->
<script src="{{ url_for('static', filename='vendor/jquery/jquery.min.js')
}}"></script>
<!-- Bootstrap Core JavaScript -->
<script src="{{ url_for('static', filename='vendor/bootstrap/js/bootstrap.min.js')
}}"></script>
<!-- Metis Menu Plugin JavaScript -->
<script src="{{ url_for('static',
filename='vendor/metisMenu/metisMenu.min.js')}}"></script>

<!-- Morris Charts JavaScript -->
<script src="{{ url_for('static',
filename='vendor/raphael/raphael.min.js')}}"></script>
```

```
_script src="{{ url_for('static', filename='data/morris-data.js' )}}"></script>

<!-- Custom Theme JavaScript -->
<script src="{{ url_for('static', filename='dist/js/sb-admin-2.js')}}"></script>
<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();

});
</script>


</body>


</html>
{% endblock %}
```

6.2 Input Screen & Output Screen

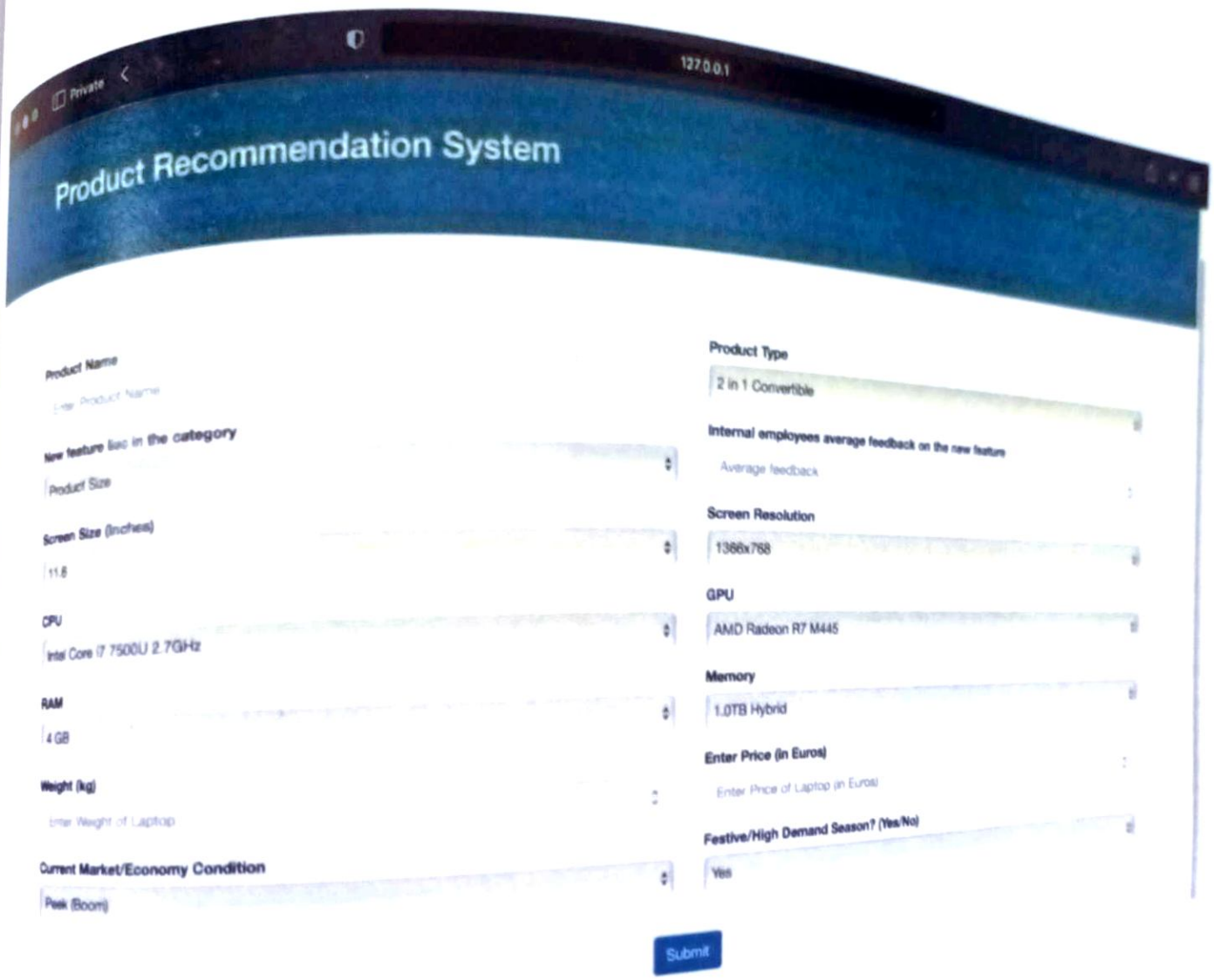


Fig. - Input Screen

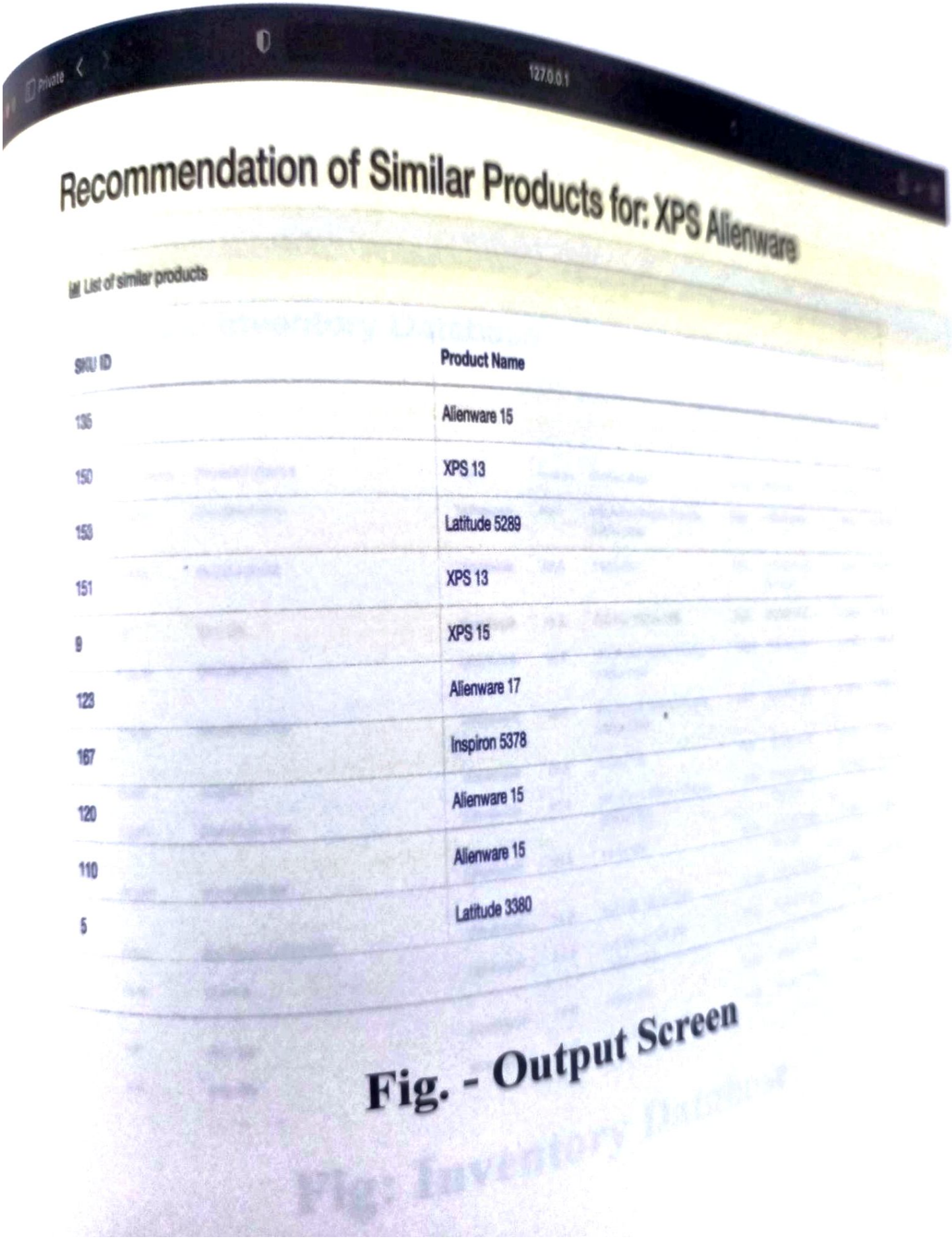


Fig. - Output Screen

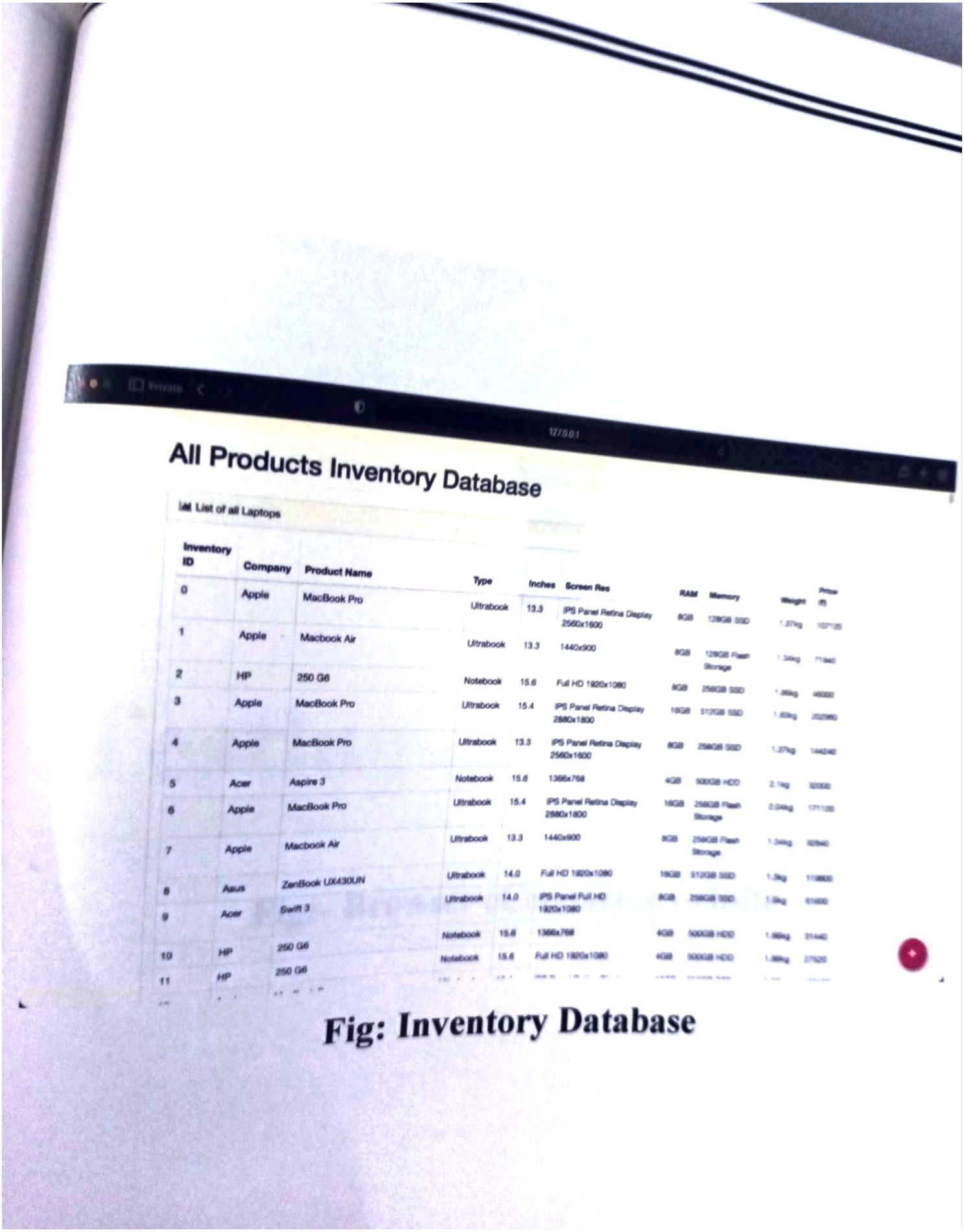


Fig: Inventory Database

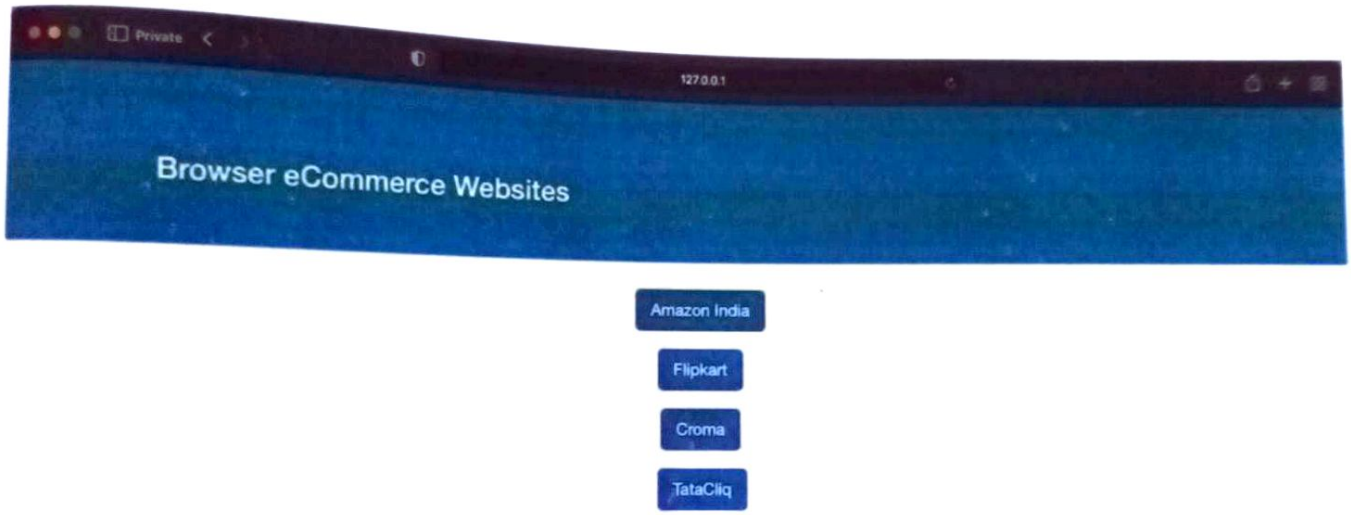


Fig:- Browser eCommerce website

Testing & Validation Checks

Warning popup when user tries to Submit without entering required information:

Fig. - Invalid input validation

Data validation popup: If user enters wrong information, system prompts with "Enter a number":

Fig. - Data validation

Implementation, evaluation and Maintenance

Implementation, evaluation and Maintenance

Implementation phase is mainly concerned with the user training, site, and preparation and file conversion. It also involves final testing of the system. During implementation the component built during development are put into optional use.

Following are the points should be considered while doing implementation of the application:

- Testing, debugging and documentation program.
- Converting data from old to new system.
- Giving training to the user about how to operate the system.
- Developing operating procedures for the computer operating staff.
- Establishing a maintenance procedure to repair and enhance system.
- Completing Documentation
- Operating system on the user location and solving all the issues
- occurred while operation.

Evaluation

After the implementation phase, another stage in project development is evaluation. After keeping the project in the working condition for sometime, all the errors that are shown in the computer program should be removed. The programmer needs to correct them so that same errors should not be repeated. We should also get the feedback from the user which are using it and ask them whether, it is user friendly or not. After evaluating the program and satisfying the needs of the user the program is maintained fully to give the same [illegible] ded to be. This stage should be implemented [illegible] dling techniques. This stage is [illegible] ing condition or

Maintenance

Maintenance is very crucial for success of any application; proper maintenance of the application makes it smooth working application. Maintenance is done basically, for two reasons i.e., to correct software errors which occurs after the testing and implementation of the application when one user it and other reason is to enhance the software capabilities in response to changing organizational needs. User often requires additional feature after he/she uses the application and becomes familiar with it. Some of the large companies gives AMC (Annual Maintenance Contract) to other companies for regular maintenance of the software/application. The cost of the maintenance increases the cost of the application/software. At a point of time, it becomes feasible to perform the tasks related to the maintenance of the software. Maintenance phase always occurs after the implementation of the application is done. It corrects all the previously undetected errors of the application and helps to do update in the application which is required by the user. Maintenance is one of the stages in the SDLC (System Development Life Cycle). It is basically done for the estimation, controlling, and making modification to the implemented system.

Future scope of the project

Future scope of the project

- Add more devices other than laptops
- Algorithm can be used for video, song recommendation.

Conclusion

Conclusion

With the help of this system a person can get information about the product in one place. This system also suggests the Product on the given requirements. Suggest similar laptops so that person can easily compare different laptops.

Bibliography

&

References

Bibliography & References

While developing this project internet was an eternal support.

Following are the websites referred by us which helped us in developing our project:

http://www.google.com

http://www.wikipedia.com

http://www.w3school.com

http://www.youtube.com