**SKILL AND JOB RECOMMENDER APPLICATION**

**A PROJECT REPORT**

***Submitted by***

**BHUVANESWARI M SIVANISREE R KANTHIMATHI I ALHASEENA KM**

**Team ID: PNT2022TMID20025**

# in partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**NATIONAL ENGINEERING COLLEGE**

(Autonomous)

**KOVILPATTI - 628503**

**ANNA UNIVERSITY: CHENNAI 600 025**

**NOVEMBER - 2022**

**NATIONAL ENGINEERING COLLEGE**

(Autonomous)

**KOVILPATTI - 628503**

**ANNA UNIVERSITY: CHENNAI 600 025 BONAFIDE CERTIFICATE**

Certified that this project report **“Skill and Job Recommender Application”** is the bonafide work of **“BHUVANESWARI M (1912005), SIVANISREE R (1912047), KANTHIMATHI I (1912013), ALHASEENA KM (1912051)”** who carried out the project work under my Mentor.

**SIGNATURE**

**Dr. V. GOMATHI. M.Tech., Ph.D.,**

**Professor & Head,**

Department of CSE,

National Engineering College (An Autonomous Institution), K.R.Nagar, Kovilpatti - 628503.

**SIGNATURE MENTOR**,

**Mr.S. Dheenathayalan, M.E., Guide/Supervisor**

**Assistant Professor,**

Department of CSE,

National Engineering College

(An Autonomous Institution),

K.R.Nagar, Kovilpatti - 628503.

Submitted for the Project viva-voce held on

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

We feel highly honoured to extend our sincere gratitude to our beloved Chairman, Lion Dr.K.S.RANGASAMY, MJF, K.S.R Educational Institutions and our Managing Trustee Mr. R.SRINIVASAN, B.B.M, MISTE, Aarthi Educational and Chairman Trust for providing all facilities to complete this project work.

We would like to acknowledge the constant and kind support provided by our highly Principal Dr. P.SENTHIL KUMAR, Ph.D.we feel highly elated to thank our respectable Head of the Department Dr. G.SINGARAVEL, M.E., Ph.D. and our project coordinator Dr. G.SINGARAVEL, M.E., Ph.D. who guided us and was a pillar of support for the successful completion of the project.

We are the most fortunate in having the opportunity to work under the Mentor Mr.Gowtham M.E CSE. express our sincere thanks to him.

Finally, I wish to thank all teaching and non-teaching staff members of the Department of Information Technology and my friends for their support towards the successful completion of this project.

I wish to acknowledge the help received from various departments and various individuals who are support to do my project in a successful manner.

SOUNDAR M SASIDHARAN S

RAMESHKRISHNAN M NISHANTH R

# 1. INTRODUCTION

* 1. **PROJECT OVERVIEW**

A system should be developed for the people who are in search of jobs with esteemed talent might get undesirable jobs and in need of bigger opportunities. Moreover, searching jobs in outer areas give complications which leads to time consuming and health issues. Hence users are in need of quicker good jobs .Meanwhile person might get jobs which are not related to their search leads to time wastage.

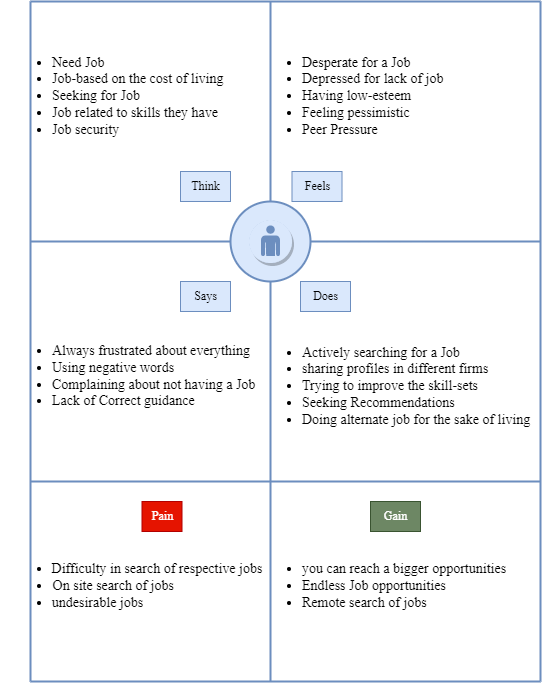
Aim is to come up with a web application through which any person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job remotely. Taking the skills from LinkedIn and throwing the best jobs according to your skillset will ease the work of candidates to get their desired jobs

# PURPOSE

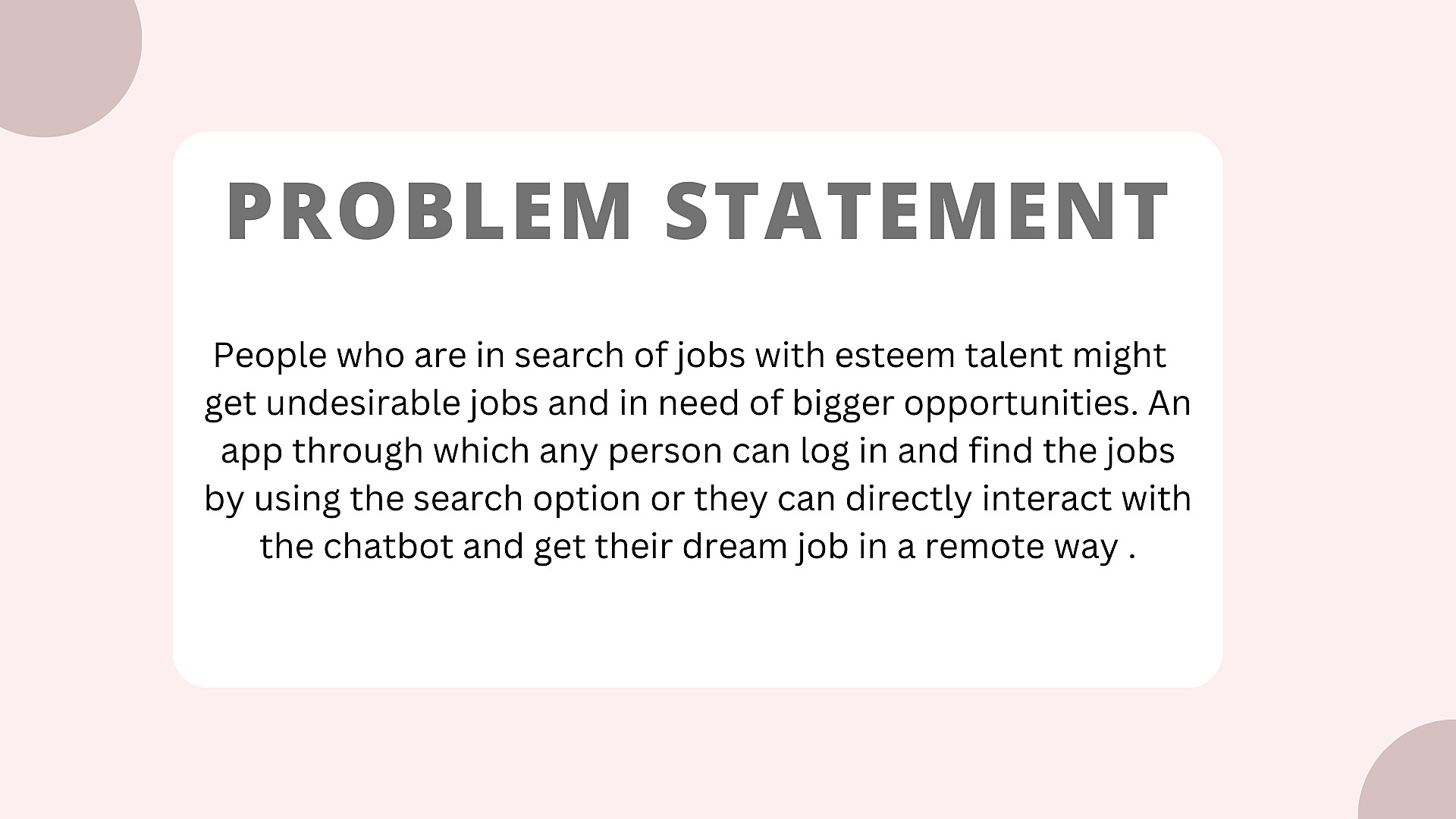
* Cost not needed for using the app
* Finding Jobs based on available domains
* Filtering Option to ease the search
* Giving user guide for the users
* Easy Interaction through chatbot.

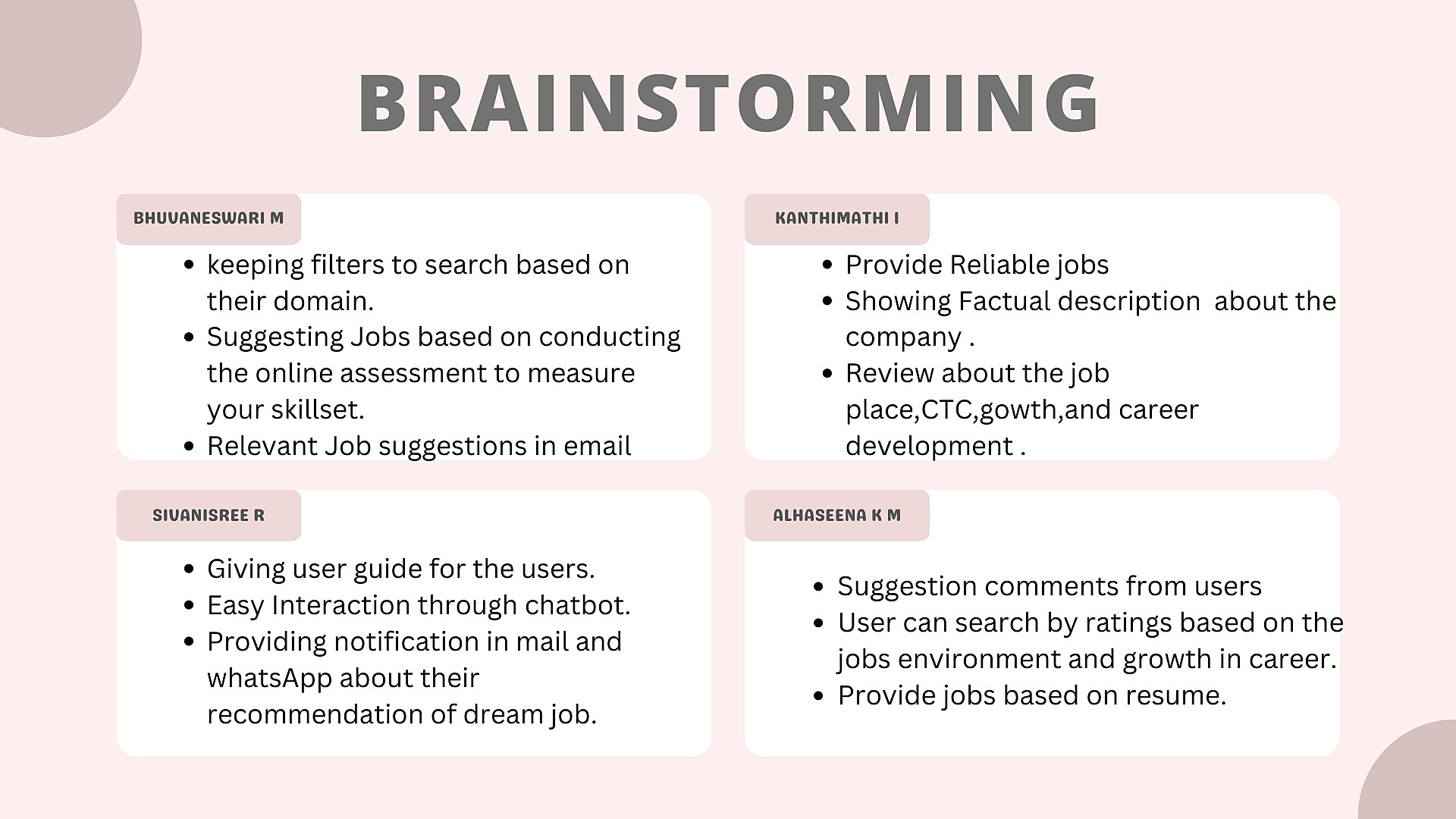
# IDEATION & PROPOSED SOLUTION

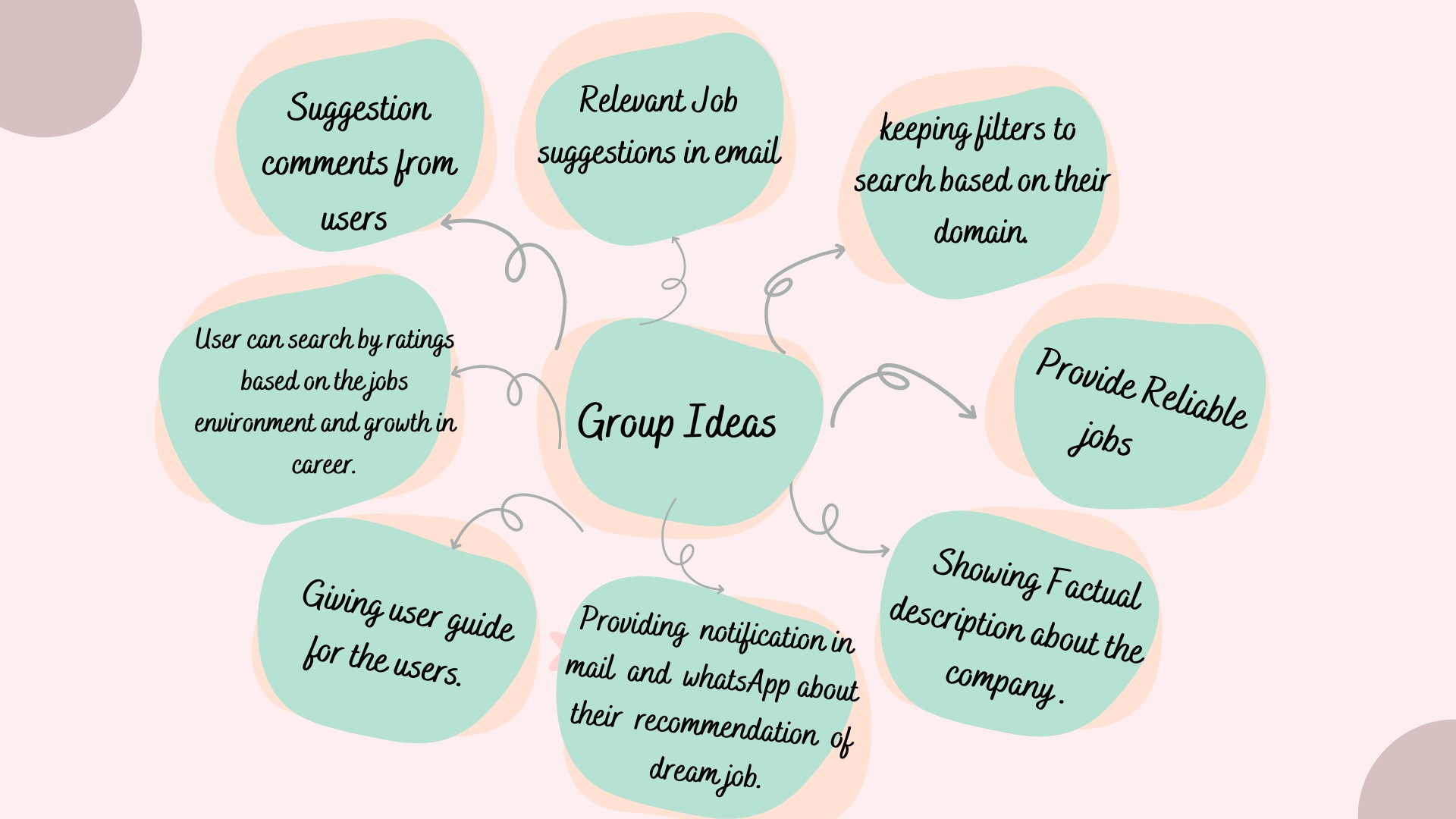
* 1. **Empathy Map Canvas**

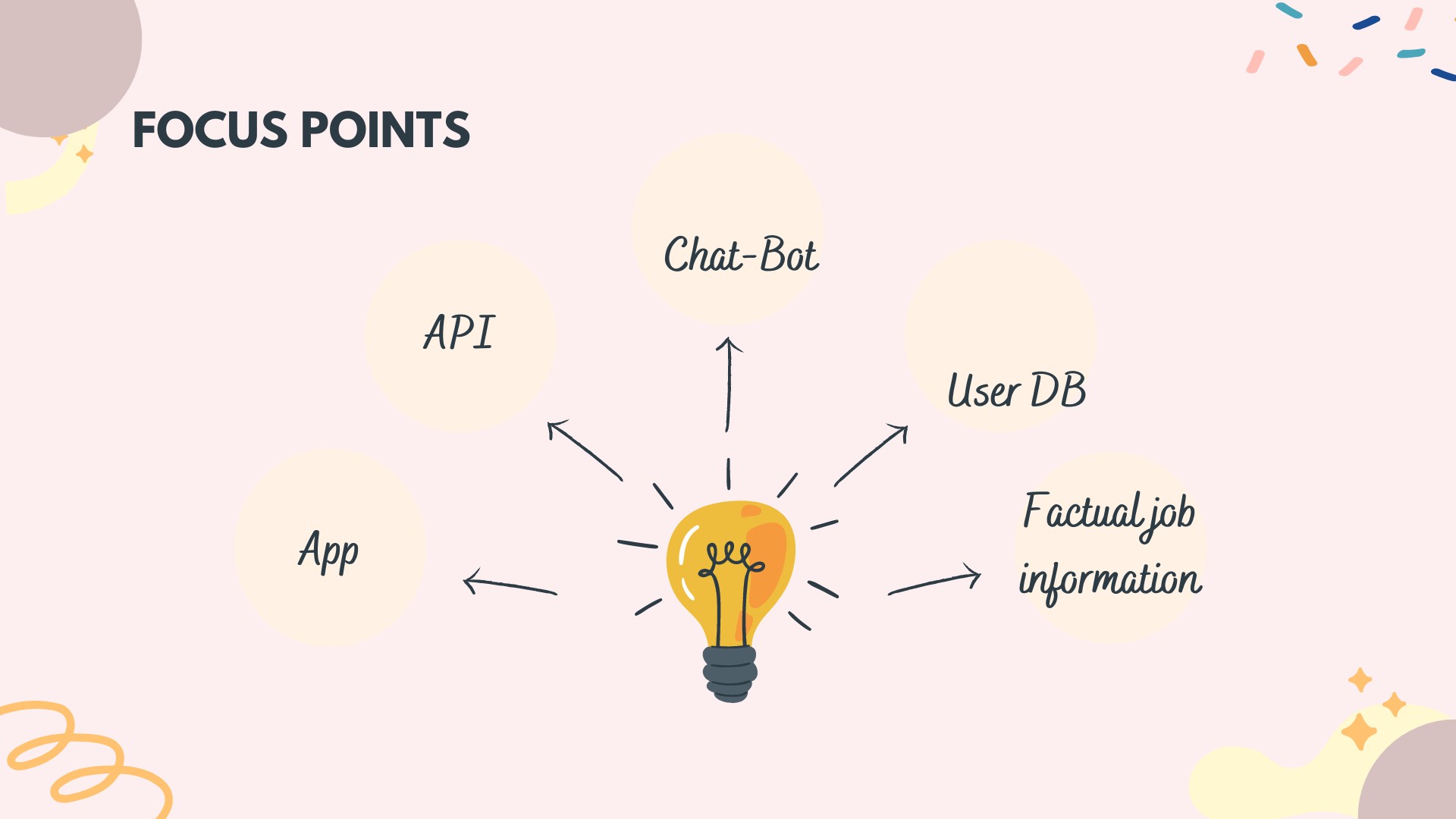


* 1. **Ideation & Brainstorming**









# Proposed Solution

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | A system should be developed for the people who are in search of jobs with esteemed talent might get undesirable jobs and in need of bigger opportunities. Moreover, searching jobs in outer areas give complications which leads to time consuming and health issues. Hence users are in need of quicker good jobs .Meanwhile person might get jobs which are not related  to their search leads to time wastage. |
| 2. | Idea / Solution description | Aim is to come up with a web application through which any person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job remotely. Taking the skills from LinkedIn and throwing the best jobs according to your skillset will ease the work of candidates to get their desired jobs. |
| 3. | Novelty / Uniqueness | * Cost not needed for using the app * Finding Jobs based on available domains * Filtering Option to ease the search * Giving user guide for the users * Easy Interaction through chatbot. |
| 4. | Social Impact / Customer Satisfaction | * Customers/Users are able to find their required jobs remotely and desirable jobs in a quick way without any distraction. |
| 5. | Business Model (Revenue Model) | This system is used when the application is created in order to achieve personal satisfaction. Its purpose is to satisfy the user, rather than making a profit from it. Many of these mobile applications can be found in the android market, since it is the most used operating system for beginners in the  programming world. |

|  |  |  |
| --- | --- | --- |
| 6. | Scalability of the Solution | * User can search by ratings based on the job’s environment and growth in career * Provide Reliable jobs * Showing Factual description about the company. * Review about the job place, CTC, growth, and career development. |

* 1. **Problem Solution Fit**

Idhula screenshot poda mudiala

# REQUIREMENT ANALYSIS

* 1. **Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | 1. Registration through Form 2. Registration through Gmail 3. Registration through LinkedIN |
| FR-2 | User Confirmation | 1. Confirmation via Email 2. Confirmation via OTP |
| FR-3 | Job Vacancies | Display job profiles based on their mentioned skills in linkedin. |
| FR-4 | Chatbot Interaction | User can solve their issues, queries and  recommend for user guides. |
| FR-5 | Improve User Profiles | User can take their interested domains test and improve their profile strength which adds advantage for hiring. |
| FR-6 | Job application and recommendations | User can apply their required jobs from their personalized jobs list and also they can get information about their recommended jobs  through mail. |

# Non-functional Requirements:

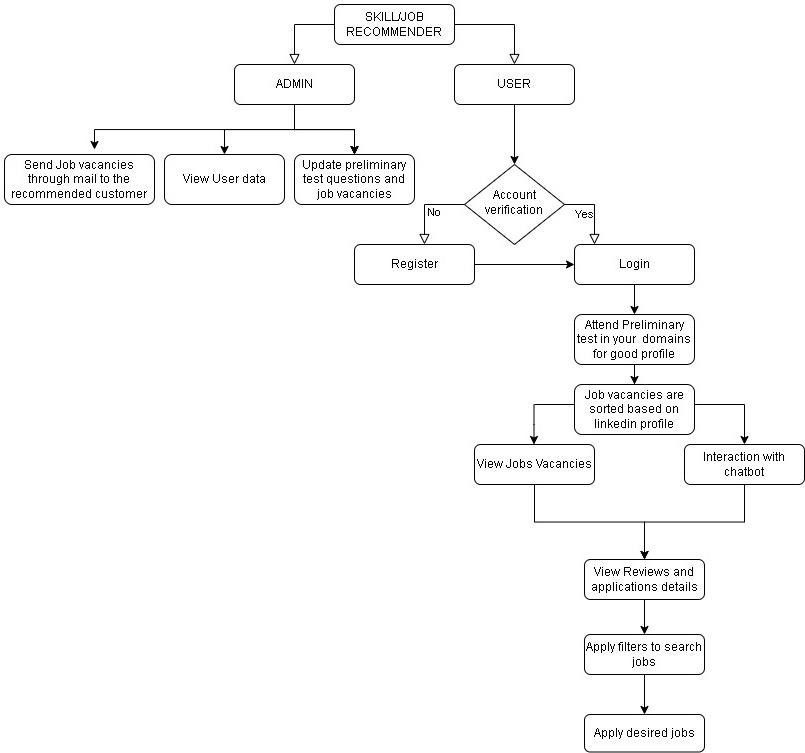
Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | 1. User can directly interact through chat bot to solve their queries. 2. User can apply jobs from their personalized jobs list in the web page. |
| NFR-2 | **Security** | 1. User data would be safely stored in IBM db2. 2. Using of SSL certificate (Python Flask   to Cloud connect) will provide security to the project. |
| NFR-3 | **Reliability** | 1. Personalized job vacancies would be listed in the web page will be always reliable when the person logged in. 2. The status of the web page will be always reliable.There won't be server   error . |
| NFR-4 | **Performance** | 1. Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic. 2. Carry out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different   configurations, within the proposed framework |
| NFR-5 | **Availability** | 1. The scraper is set up to avoid duplicate job offers, thus all the job offers are unique. 2. To making the user reliable. This   webpage will be available to all users (network connectivity is necessary) at any given point of time.   1. Made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected   from different job search engine sites. |
| NFR-6 | **Scalability** | 1. Hiring people can take their candidates unlimitedly. 2. Unlimited personalized jobs will be   listed on the web page based on their  skill. |

# PROJECT DESIGN

* 1. **Data Flow Diagrams**

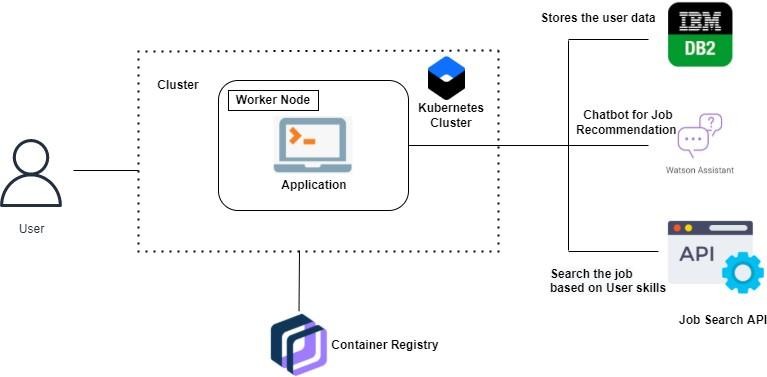
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



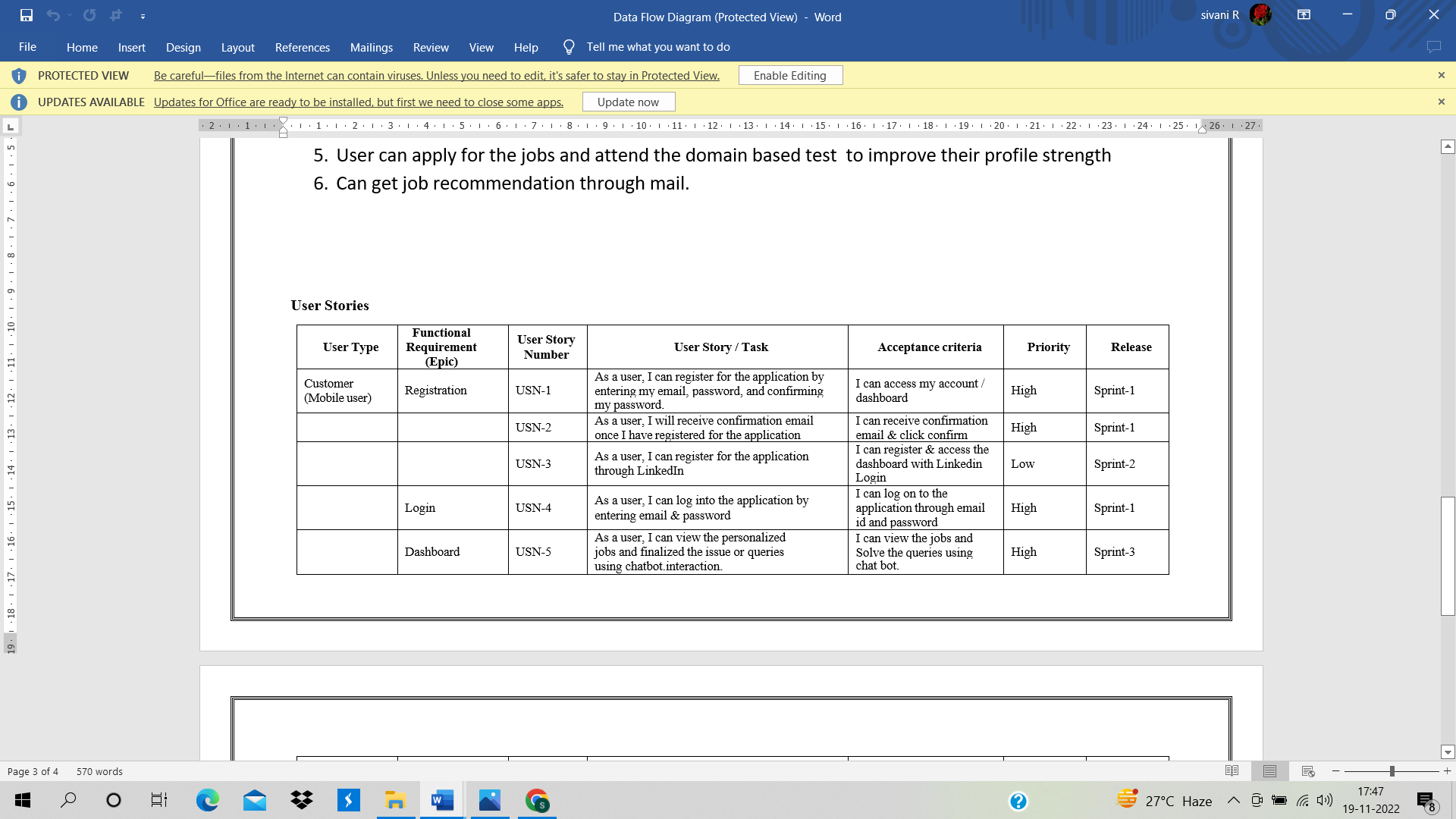
# FLOW:

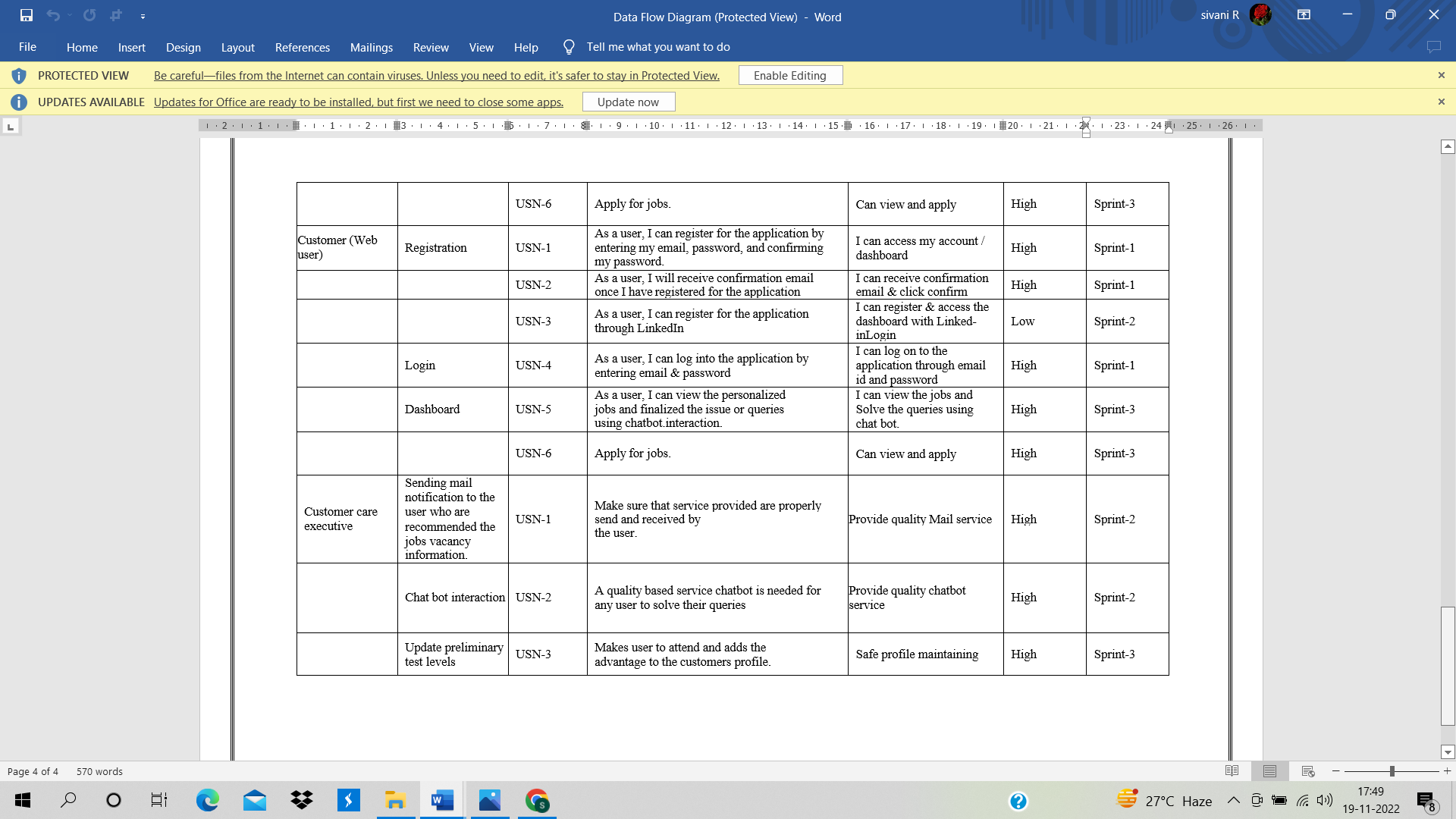
1. User starts the app by finishing the login credentials.
2. Based on the skill mentioned in the LinkedIn the personalized jobs will be viewed by the user.
3. User who are hiring the candidates will also be able to view the candidate profile.
4. Interact with chatbot used for clearing the queries.
5. User can apply for the jobs and attend the domain-based test to improve their profile strength
6. Can get job recommendation through mail.

# SOLUTION & TECHNICAL ARCHITECTURE



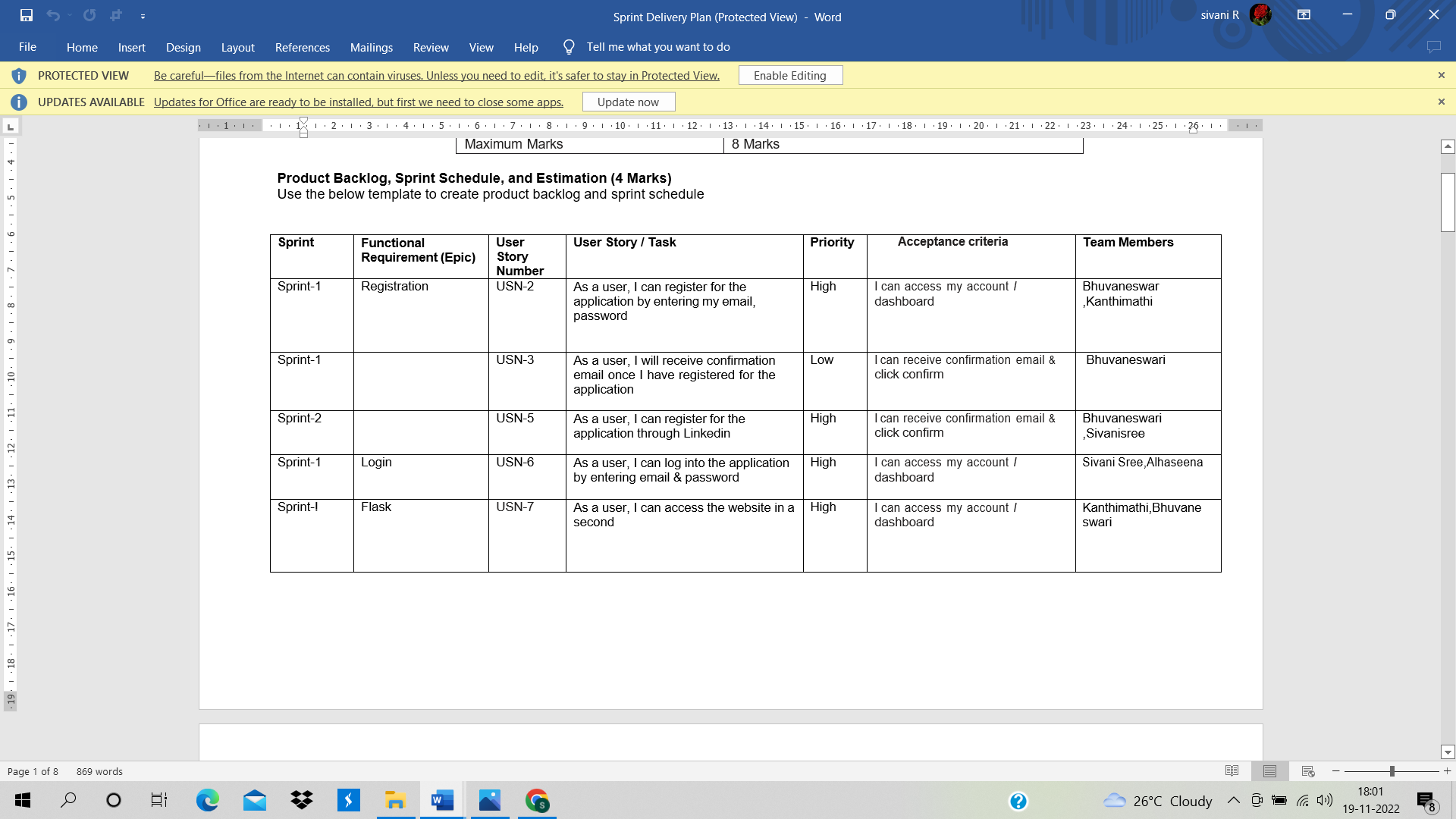
* 1. **USER STORIES**

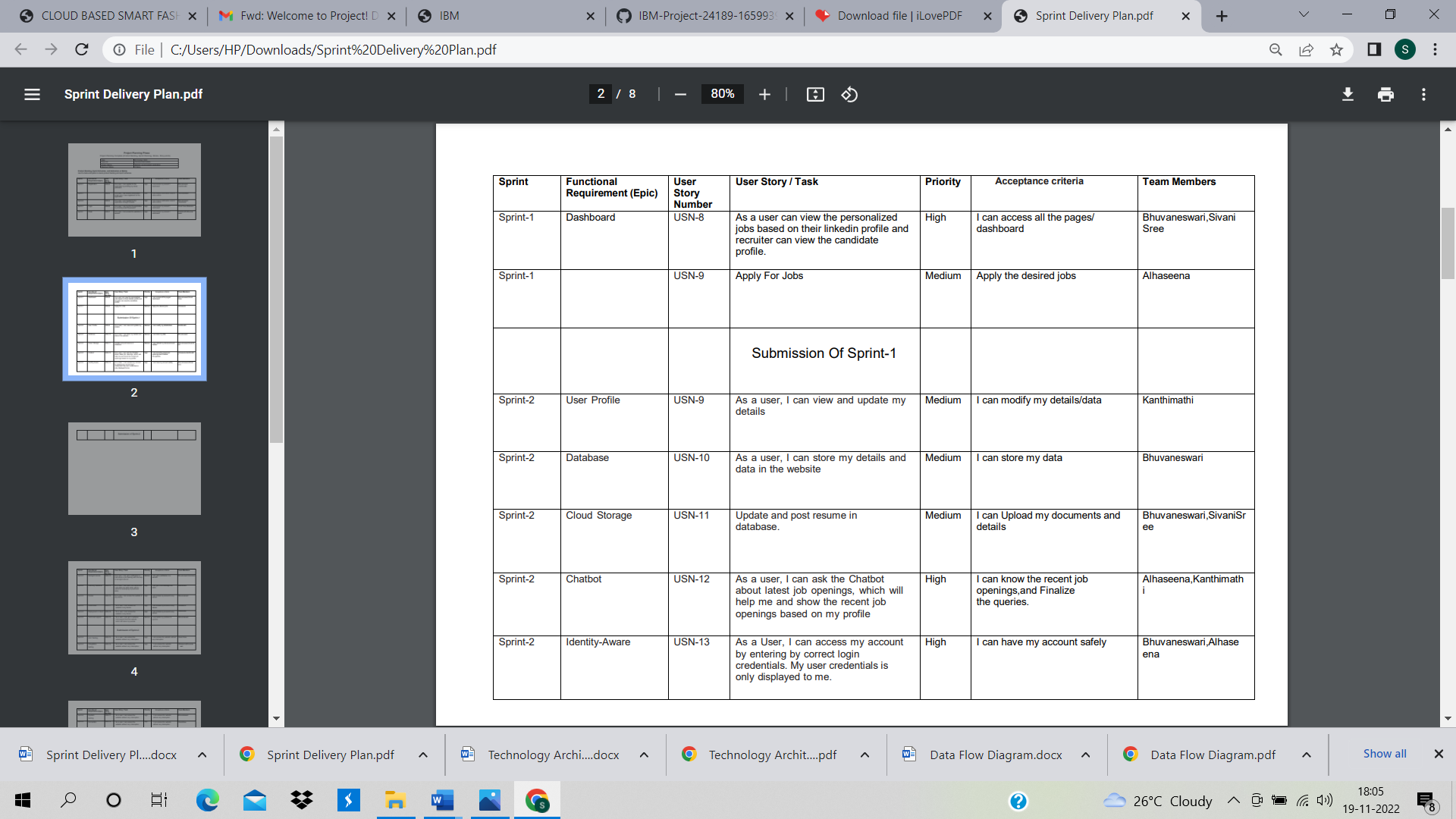


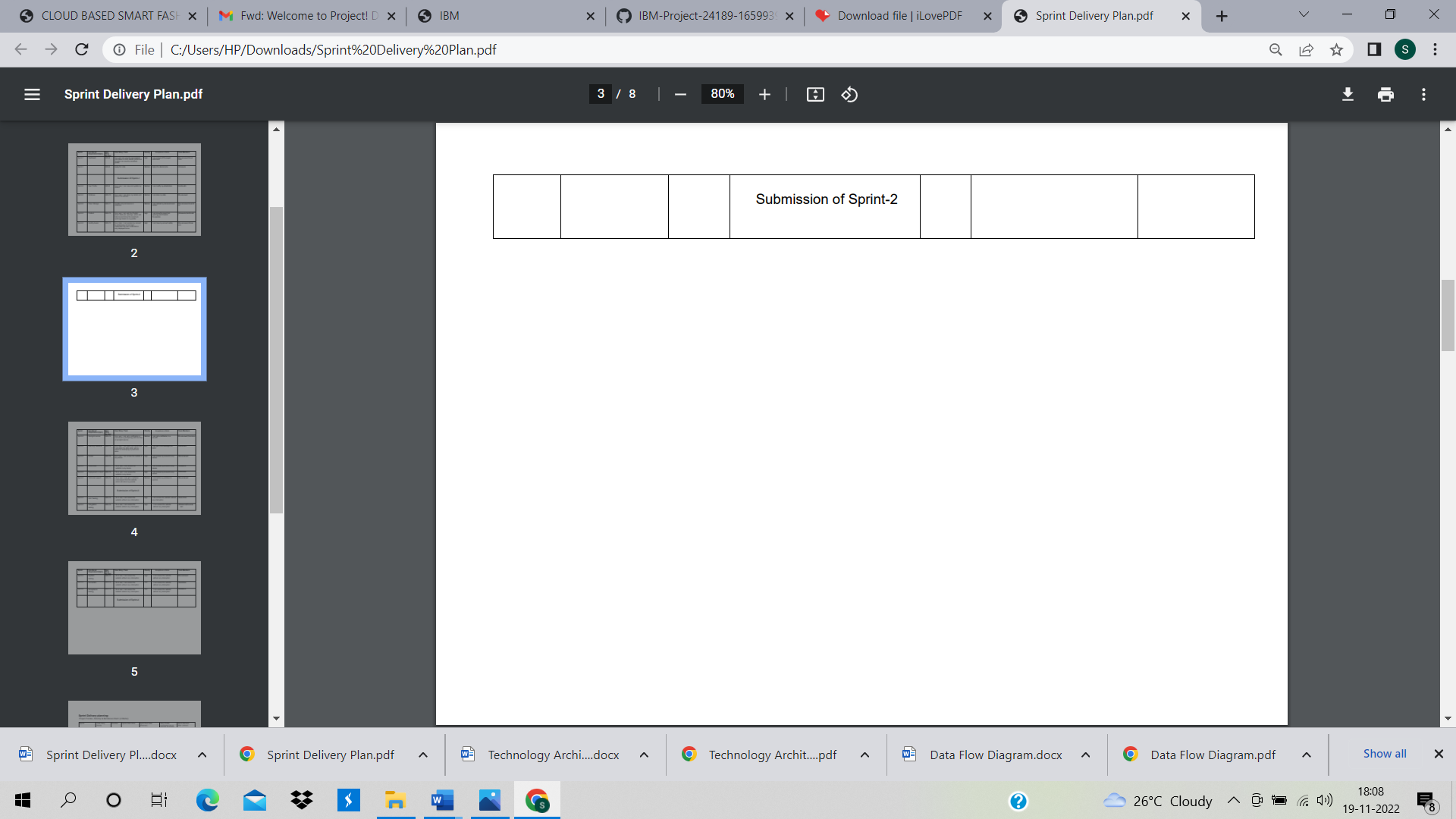


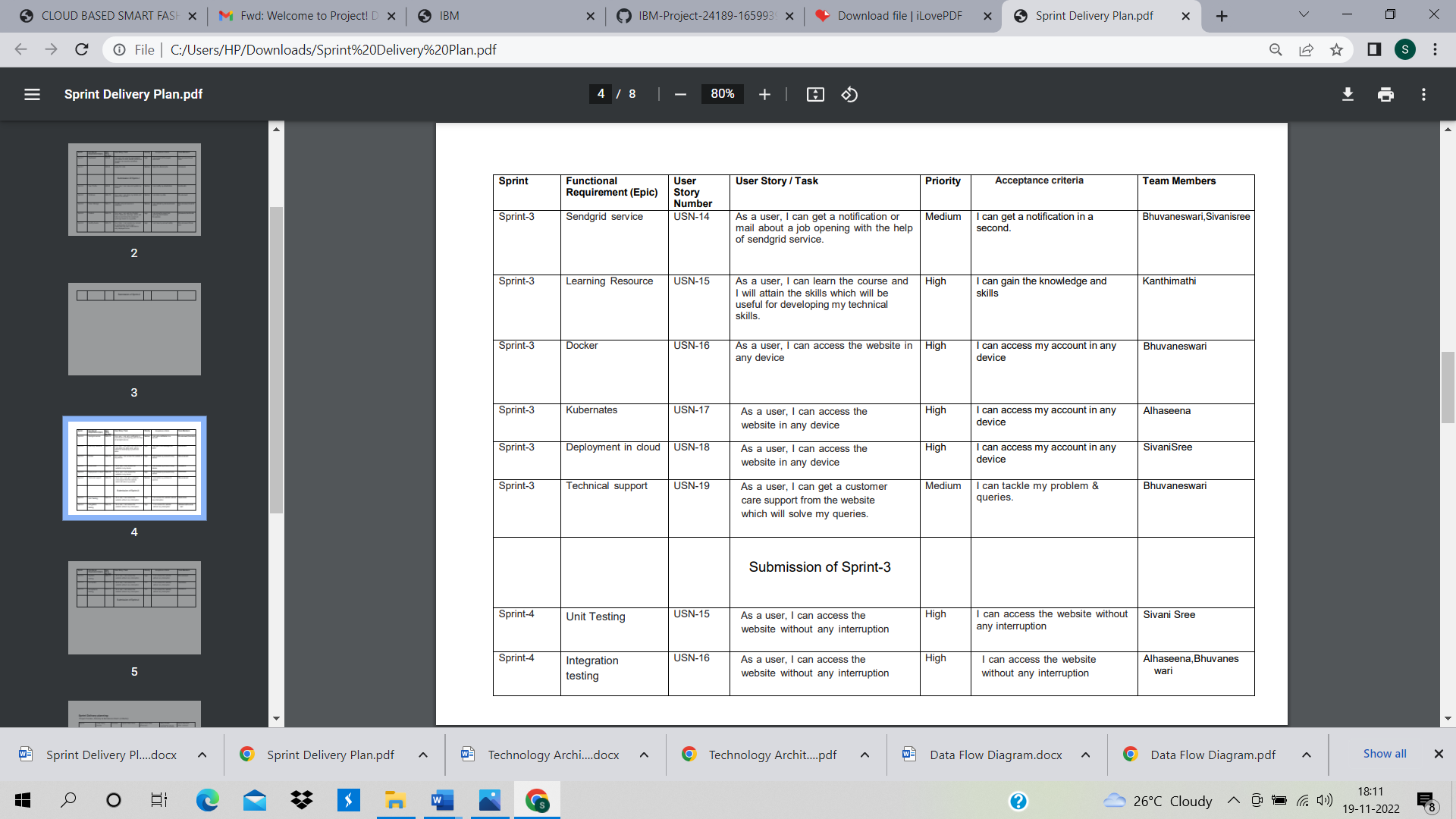
# PROJECT PLANNING & SCHEDULING

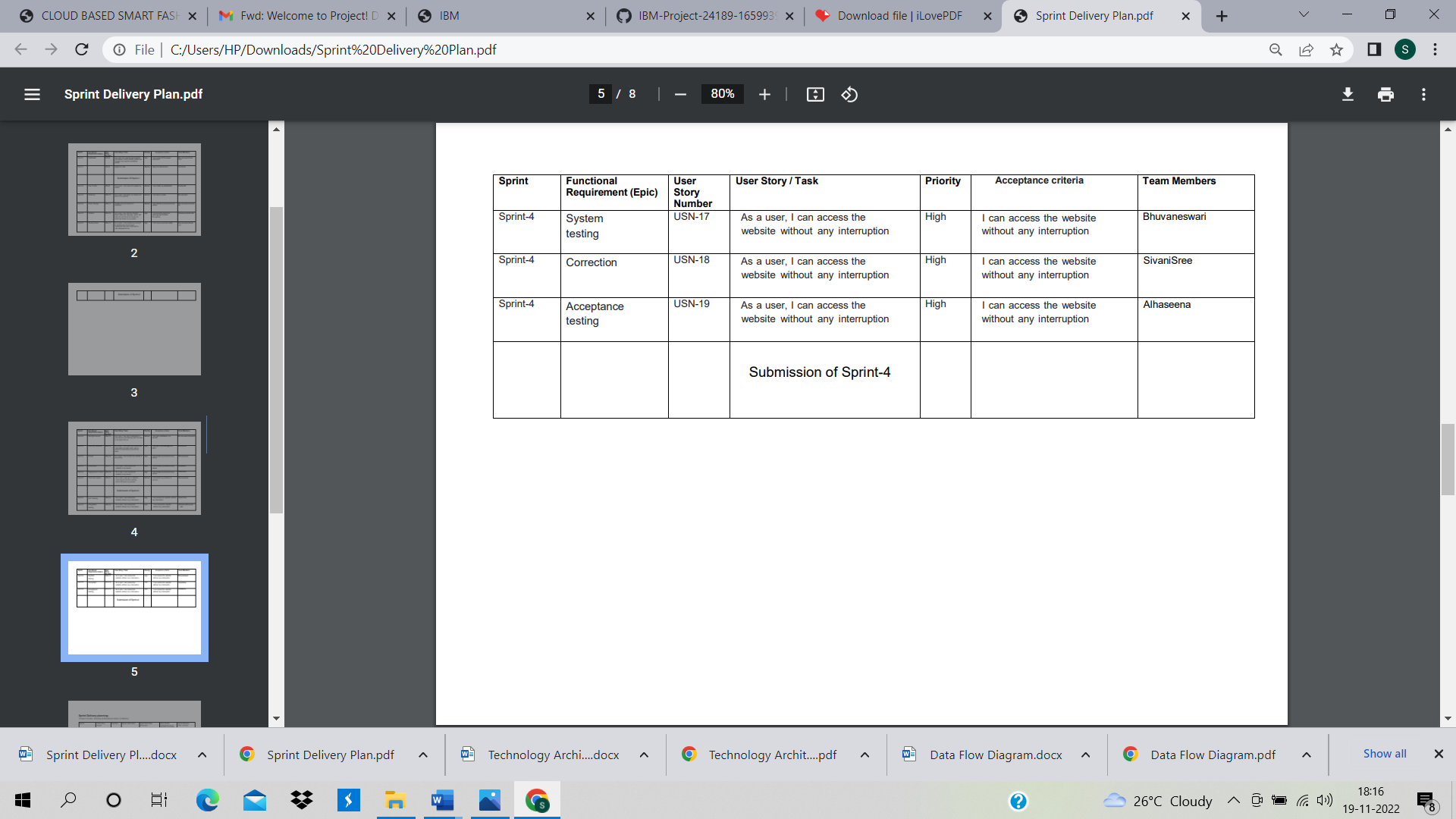
* 1. **SPRINT PLANNING & ESTIMATION**











# SPRINT DELIEVERY SCHEDULE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

**Velocity:**

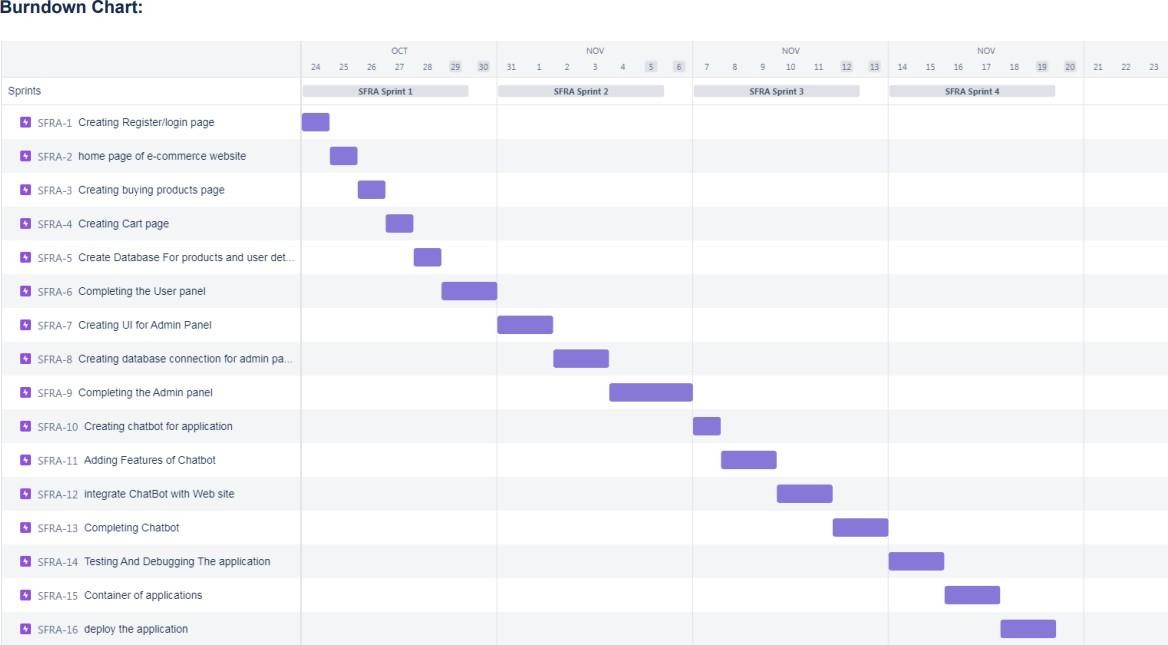
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

*sprint duration* 20

2

*AV*

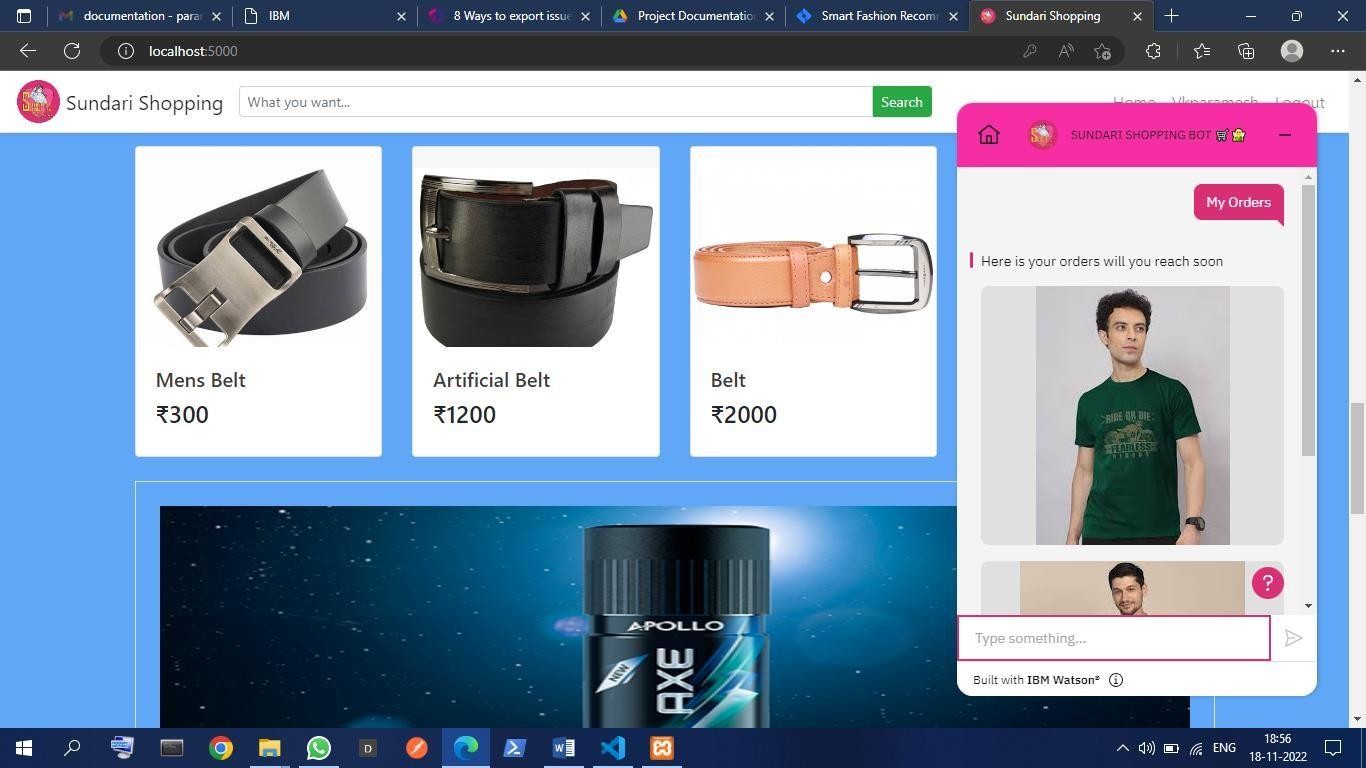
*velocity* 10

* 1. **REPORTS FROM JIRA**

# CODING & SOLUTIONING

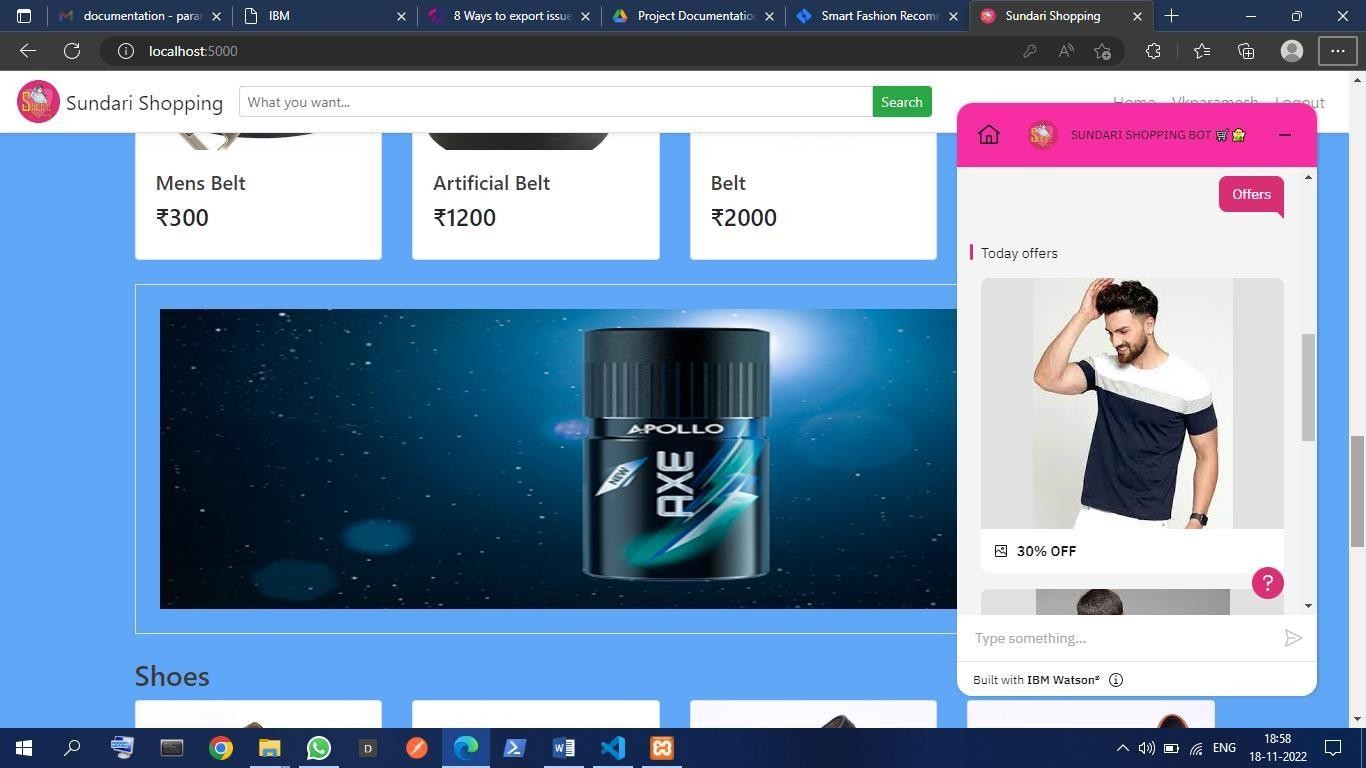
* 1. **Feature 1**

Using chat bot we can manage user's choices and orders.

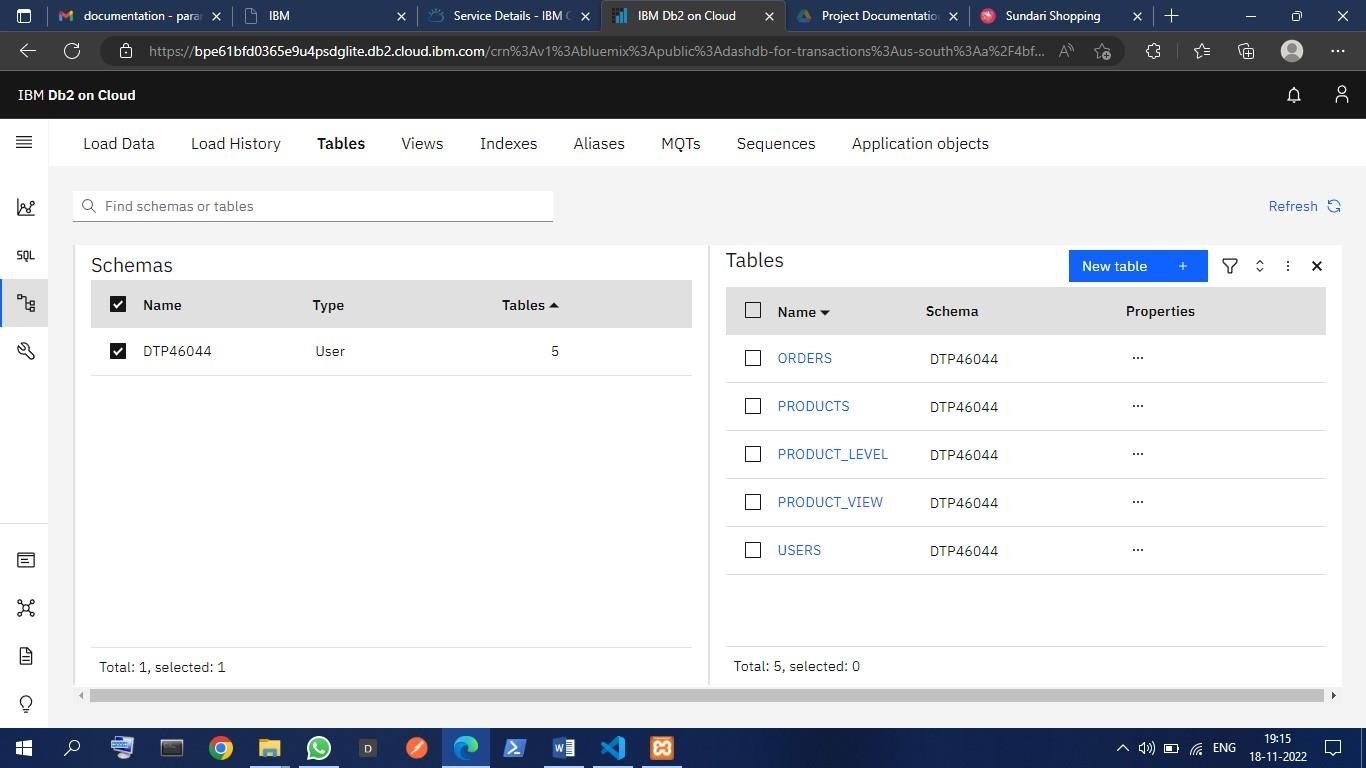


# Feature 2

Chat Bot promote the best deals and offers on that day.



# Database Schema



1. **TESTING**

# Test Cases

This report shows the number of test cases that have passed, failed, and untested

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Login | 5 | 0 | 0 | 5 |
| Register | 7 | 0 | 0 | 7 |
| Home Page | 2 | 0 | 0 | 2 |
| Order page | 3 | 0 | 0 | 3 |
| Order products | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

# USER ACCEPTANCE TESTING

**Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Fashion Recommender Application project at the time of the release to User Acceptance Testing (UAT).

# Defect Analysis

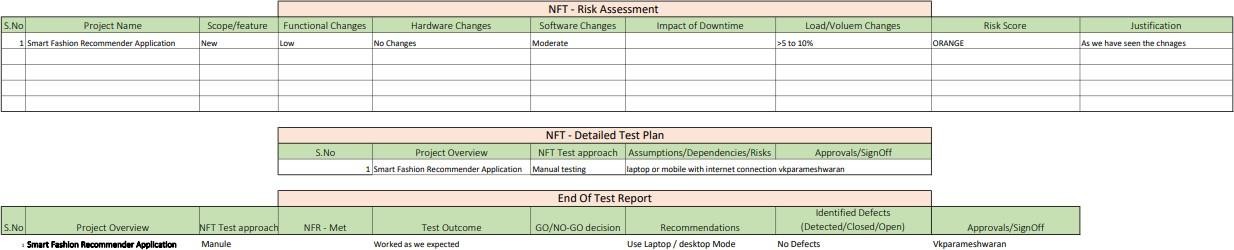
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 5 | 5 | 2 | 3 | 21 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 24 | 14 | 13 | 26 | 77 |

# RESULT

* 1. **PERFORMANCE METRICS**

Project team shall fill the following information in model performance testing.



# ADVANTAGES & DISADVANTAGES

**ADVANTAGES:**

* + - Its helps to user Shopping with Assistant
    - Its helps to user manage there order list
    - Its helps to user shopping at home

# DISADVANTAGES:

* + - User have fear about online shopping
    - User have sometimes receives wrong items
    - User have fear about online payment

# CONCLUSION

Recommendation systems have the potential to explore new opportunities for retailers by enabling them to provide customized recommendations to consumers based on information retrieved from the Internet. They help consumers to instantly find the products and services that closely match with their choices. Moreover, different stat-of-the-art algorithms have been developed to recommend products based on users’ interactions with their social groups. Therefore, research on embedding social media images within fashion recommendation systems has gained huge popularity in recent times. This paper presented a review of the fashion recommendation systems, algorithmic models and filtering techniques based on the academic articles related to this topic. The technical aspects, strengths and weaknesses of the filtering techniques have been discussed elaborately, which will help future researchers gain an in-depth understanding of fashion recommender systems. However, the proposed prototypes should be tested in commercial applications to understand their feasibility and accuracy in the retail market, because inaccurate recommendations can produce a negative impact on a customer. Moreover, future research should concentrate on including time series analysis and accurate categorization of product images based on the variation in color, trend and clothing style in order to develop an effective recommendation system. The proposed model will follow brand specific personalization campaigns and hence it will ensure highly curated and tailored offerings for users. Hence, this research will be highly beneficial for researchers interested in using augmented and virtual reality features to develop recommendation systems.

# FUTURE SCOPE

There has been significant progress recently in fashion recommendation system research, which will benefit both consumers and retailers soon. The use of product and user images, textual content, demographic history, and cultural information is crucial in developing recommendation frameworks. Product attributes and clothing style matching are common features of collaborative and content-based filtering techniques. Researchers can develop more sophisticated hyper personalized filtering techniques considering the correlation between consumers’ clothing styles and personalities. The methods based on employing a scoring system for quantifying each product attribute will be helpful in increasing the precision of the model. The use of virtual sales advisers in an online shopping portal would provide consumers with a real time offline shopping experience. Retailers can collect the data on users’ purchase history and product reviews from the recommendation system and subsequently use them in style prediction for the upcoming seasons. The integration of different domain information strengthens the deep learning paradigm by enabling the detection of design component variation, which improves the performance of the recommendation system in the long run. Deep learning approaches should be more frequently used to quickly explore fashion items from different online databases to provide prompt recommendations to users or consumers.

# 12. APPENDIX

1. **Source Code App.py**

*from* flask *import* Flask, render\_template, flash, redirect, url\_for, session, request, logging

*from* sqlalchemy *import* sqlalchemy

*from* wtforms *import* Form, StringField, TextAreaField, PasswordField, validators, SelectField

*from* passlib.hash *import* sha256\_crypt

*from* functools *import* wraps

*from* flask\_uploads *import* UploadSet, configure\_uploads, IMAGES

*import* timeit

*import* datetime

*from* flask\_mail *import* Mail, Message

*import* os

*from* wtforms *import* EmailField

*import* sendgrid

*import* os

db = sqlalchemy.create\_engine('ibm\_db\_sa://dtp46044:95soX0sZGhb4ToUj@fbd88901-ebdb-4a4f-a32e- 9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:50000/bludb')

bcrypt =Bcrypt(app) metadata = MetaData()

app = Flask( name ) app.secret\_key = os.urandom(24)

app.config['UPLOADED\_PHOTOS\_DEST'] = 'static/image/product' photos = UploadSet('photos', IMAGES)

configure\_uploads(app, photos)

db.init\_app(app) def *sendemail*():

sg = sendgrid.SendGridAPIClient(api\_key=os.environ.get('SENDGRID\_API\_KEY')) data = {

"personalizations": [

{

"to": [

{

"email": "[vkdj3232@gmail.com](mailto:vkdj3232@gmail.com)"

}

],

"subject": "Sundari Shopping.Thankyou for purchasing"

}

],

"from": {

"email": "[parameshvk3232@gmail.com](mailto:parameshvk3232@gmail.com)"

},

"content": [

{

"type": "text/plain",

"value": "Oder successfull your order arive within 5 working days"

}

]

}

response = sg.client.mail.send.post(request\_body=data) print(response.status\_code)

print(response.body) print(response.headers)

def *is\_logged\_in*(f): @*wraps*(f)

def *wrap*(\*args, \*\*kwargs):

*if* 'logged\_in' in session:

*return* f(\*args, \*kwargs)

*else*:

*return* redirect(url\_for('login'))

*return* wrap

def *not\_logged\_in*(f): @*wraps*(f)

def *wrap*(\*args, \*\*kwargs):

*if* 'logged\_in' in session:

*return* redirect(url\_for('index'))

*else*:

*return* f(\*args, \*kwargs)

*return* wrap

def *is\_admin\_logged\_in*(f): @*wraps*(f)

def *wrap*(\*args, \*\*kwargs):

*if* 'admin\_logged\_in' in session:

*return* f(\*args, \*kwargs)

*else*:

*return* redirect(url\_for('admin\_login'))

*return* wrap

def *not\_admin\_logged\_in*(f): @*wraps*(f)

def *wrap*(\*args, \*\*kwargs):

*if* 'admin\_logged\_in' in session:

*return* redirect(url\_for('admin'))

*else*:

*return* f(\*args, \*kwargs)

*return* wrap

def *wrappers*(func, \*args, \*\*kwargs): def *wrapped*():

*return* func(\*args, \*\*kwargs)

*return* wrapped

def *content\_based\_filtering*(product\_id): cur = db.connection.cursor()

cur.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) # *getting id row*

data = cur.fetchone() # *get row info*

data\_cat = data['category'] # *get id category ex shirt*

print('Showing result for Product Id: ' + product\_id)

category\_matched = cur.execute("SELECT \* FROM products WHERE category=%s", (data\_cat,)) # *get all shirt category*

print('Total product matched: ' + str(category\_matched)) cat\_product = cur.fetchall() # *get all row*

cur.execute("SELECT \* FROM product\_level WHERE product\_id=%s", (product\_id,)) # *id level info*

id\_level = cur.fetchone() recommend\_id = []

cate\_level = ['v\_shape', 'polo', 'clean\_text', 'design', 'leather', 'color', 'formal', 'converse', 'loafer', 'hook', 'chain']

*for* product\_f *in* cat\_product:

cur.execute("SELECT \* FROM product\_level WHERE product\_id=%s", (product\_f['id'],)) f\_level = cur.fetchone()

match\_score = 0

*if* f\_level['product\_id'] != int(product\_id):

*for* cat\_level *in* cate\_level:

*if* f\_level[cat\_level] == id\_level[cat\_level]: match\_score += 1

*if* match\_score == 11: recommend\_id.append(f\_level['product\_id'])

print('Total recommendation found: ' + str(recommend\_id))

*if* recommend\_id:

cur = db.connection.cursor()

placeholders = ','.join((str(n) *for* n *in* recommend\_id))

query = 'SELECT \* FROM products WHERE id IN (%s)' % placeholders cur.execute(query)

recommend\_list = cur.fetchall()

*return* recommend\_list, recommend\_id, category\_matched, product\_id

*else*:

*return* ''

@*app*.*route*('/') @*is\_logged\_in* def *index*():

form = OrderForm(request.form) # *Create cursor*

cur = db.connection.cursor() # *Get message*

values = 'tshirt'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,)) tshirt = cur.fetchall()

values = 'wallet'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,)) wallet = cur.fetchall()

values = 'belt'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,)) belt = cur.fetchall()

values = 'shoes'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY RAND() LIMIT 4", (values,)) shoes = cur.fetchall()

# *Close Connection*

cur.close()

*return* render\_template('home.html', tshirt=tshirt, wallet=wallet, belt=belt, shoes=shoes, form=form)

class LoginForm(Form): # *Create Login Form*

username = StringField('', [validators.length(min=1)], render\_kw={'autofocus': True, 'placeholder': 'Username'})

password = PasswordField('', [validators.length(min=3)],

render\_kw={'placeholder': 'Password'})

# *User Login*

@*app*.*route*('/login', methods=['GET', 'POST']) @*not\_logged\_in*

def *login*():

form = LoginForm(request.form)

*if* request.method == 'POST' and form.validate(): # *GEt user form*

username = form.username.data

# *password\_candidate = request.form['password']*

password\_candidate = form.password.data

# *Create cursor*

cur = db.connection.cursor()

# *Get user by username*

result = cur.execute("SELECT \* FROM users WHERE username=%s", [username])

*if* result > 0:

# *Get stored value*

data = cur.fetchone() password = data['password'] uid = data['id']

name = data['name']

# *Compare password*

*if* sha256\_crypt.verify(password\_candidate, password): # *passed*

session['logged\_in'] = True session['uid'] = uid session['s\_name'] = name x = '1'

cur.execute("UPDATE users SET online=%s WHERE id=%s", (x, uid))

*return* redirect(url\_for('index'))

*else*:

flash('Incorrect password', 'danger')

*return* render\_template('login.html', form=form)

*else*:

flash('Username not found', 'danger') # *Close connection*

cur.close()

*return* render\_template('login.html', form=form)

*return* render\_template('login.html', form=form)

@*app*.*route*('/out') def *logout*():

*if* 'uid' in session: # *Create cursor*

cur = db.connection.cursor() uid = session['uid']

x = '0'

cur.execute("UPDATE users SET online=%s WHERE id=%s", (x, uid)) session.clear()

flash('You are logged out', 'success')

*return* redirect(url\_for('index'))

*return* redirect(url\_for('login'))

class RegisterForm(Form):

name = StringField('', [validators.length(min=3, max=50)], render\_kw={'autofocus': True, 'placeholder': 'Full Name'})

username = StringField('', [validators.length(min=3, max=25)], render\_kw={'placeholder': 'Username'}) email = EmailField('', [validators.DataRequired(), validators.Email(), validators.length(min=4, max=25)],

render\_kw={'placeholder': 'Email'}) password = PasswordField('', [validators.length(min=3)],

render\_kw={'placeholder': 'Password'})

mobile = StringField('', [validators.length(min=11, max=15)], render\_kw={'placeholder': 'Mobile'})

@*app*.*route*('/register', methods=['GET', 'POST']) @*not\_logged\_in*

def *register*():

form = RegisterForm(request.form)

*if* request.method == 'POST' and form.validate(): name = form.name.data

email = form.email.data username = form.username.data

password = sha256\_crypt.encrypt(str(form.password.data)) mobile = form.mobile.data

# *insert\_sql = "INSERT INTO users VALUES (?, ?, ?)"*

# *prep\_stmt = ibm\_db.prepare(conn, insert\_sql)* # *ibm\_db.bind\_param(prep\_stmt, 1, username)* # *ibm\_db.bind\_param(prep\_stmt, 2, email)*

# *ibm\_db.bind\_param(prep\_stmt, 3, password)*

# *ibm\_db.execute(prep\_stmt)*

# *Create Cursor*

cur = db.connection.cursor()

cur.execute("INSERT INTO users(name, email, username, password, mobile) VALUES(%s, %s, %s, %s,

%s)",

(name, email, username, password, mobile))

# *Commit cursor*

db.connection.commit()

# *Close Connection*

cur.close()

flash('You are now registered and can login', 'success')

*return* redirect(url\_for('index'))

*return* render\_template('register.html', form=form)

class MessageForm(Form): # *Create Message Form*

body = StringField('', [validators.length(min=1)], render\_kw={'autofocus': True})

@*app*.*route*('/chatting/<string:id>', methods=['GET', 'POST']) def *chatting*(id):

*if* 'uid' in session:

form = MessageForm(request.form) # *Create cursor*

cur = db.connection.cursor()

# *lid name*

get\_result = cur.execute("SELECT \* FROM users WHERE id=%s", [id]) l\_data = cur.fetchone()

*if* get\_result > 0:

session['name'] = l\_data['name'] uid = session['uid'] session['lid'] = id

*if* request.method == 'POST' and form.validate(): txt\_body = form.body.data

# *Create cursor*

cur = db.connection.cursor()

cur.execute("INSERT INTO messages(body, msg\_by, msg\_to) VALUES(%s, %s, %s)", (txt\_body, id, uid))

# *Commit cursor*

db.connection.commit()

# *Get users*

cur.execute("SELECT \* FROM users") users = cur.fetchall()

# *Close Connection*

cur.close()

*return* render\_template('chat\_room.html', users=users, form=form)

*else*:

flash('No permission!', 'danger')

*return* redirect(url\_for('index'))

*else*:

*return* redirect(url\_for('login'))

@*app*.*route*('/chats', methods=['GET', 'POST']) def *chats*():

*if* 'lid' in session:

id = session['lid'] uid = session['uid'] # *Create cursor*

cur = db.connection.cursor() # *Get message*

cur.execute("SELECT \* FROM messages WHERE (msg\_by=%s AND msg\_to=%s) OR (msg\_by=%s AND msg\_to=%s) "

"ORDER BY id ASC", (id, uid, uid, id)) chats = cur.fetchall()

# *Close Connection*

cur.close()

*return* render\_template('chats.html', chats=chats, )

*return* redirect(url\_for('login'))

class OrderForm(Form): # *Create Order Form*

name = StringField('', [validators.length(min=1), validators.DataRequired()], render\_kw={'autofocus': True, 'placeholder': 'Full Name'})

mobile\_num = StringField('', [validators.length(min=1), validators.DataRequired()], render\_kw={'autofocus': True, 'placeholder': 'Mobile'})

quantity = SelectField('', [validators.DataRequired()], choices=[('1', '1'), ('2', '2'), ('3', '3'), ('4', '4'), ('5', '5')])

order\_place = StringField('', [validators.length(min=1), validators.DataRequired()], render\_kw={'placeholder': 'Order Place'})

@*app*.*route*('/tshirt', methods=['GET', 'POST']) def *tshirt*():

form = OrderForm(request.form) # *Create cursor*

cur = db.connection.cursor() # *Get message*

values = 'tshirt'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY id ASC", (values,)) products = cur.fetchall()

# *Close Connection*

cur.close()

*if* request.method == 'POST' and form.validate(): name = form.name.data

mobile = form.mobile\_num.data order\_place = form.order\_place.data quantity = form.quantity.data

pid = request.args['order']

now = datetime.datetime.now() week = datetime.timedelta(days=7) delivery\_date = now + week

now\_time = delivery\_date.strftime("%y-%m-%d %H:%M:%S") # *Create Cursor*

curs = db.connection.cursor()

*if* 'uid' in session:

uid = session['uid']

curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s, %s)",

(uid, pid, name, mobile, order\_place, quantity, now\_time))

*else*:

curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s)",

(pid, name, mobile, order\_place, quantity, now\_time)) # *Commit cursor*

db.connection.commit()

# *Close Connection* cur.close() sendemail()

flash('Order successful', 'success')

*return* render\_template('tshirt.html', tshirt=products, form=form)

*if* 'view' in request.args:

product\_id = request.args['view'] curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,))

product = curso.fetchall()

x = content\_based\_filtering(product\_id)

wrappered = wrappers(content\_based\_filtering, product\_id) execution\_time = timeit.timeit(wrappered, number=0)

# *print('Execution time: ' + str(execution\_time) + ' usec') if* 'uid' in session:

uid = session['uid'] # *Create cursor*

cur = db.connection.cursor()

cur.execute("SELECT \* FROM product\_view WHERE user\_id=%s AND product\_id=%s", (uid, product\_id))

result = cur.fetchall()

*if* result:

now = datetime.datetime.now()

now\_time = now.strftime("%y-%m-%d %H:%M:%S")

cur.execute("UPDATE product\_view SET date=%s WHERE user\_id=%s AND product\_id=%s", (now\_time, uid, product\_id))

*else*:

cur.execute("INSERT INTO product\_view(user\_id, product\_id) VALUES(%s, %s)", (uid, product\_id))

db.connection.commit()

*return* render\_template('view\_product.html', x=x, tshirts=product)

*elif* 'order' in request.args: product\_id = request.args['order'] curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) product = curso.fetchall()

x = content\_based\_filtering(product\_id)

*return* render\_template('order\_product.html', x=x, tshirts=product, form=form)

*return* render\_template('tshirt.html', tshirt=products, form=form)

@*app*.*route*('/wallet', methods=['GET', 'POST']) def *wallet*():

form = OrderForm(request.form) # *Create cursor*

cur = db.connection.cursor() # *Get message*

values = 'wallet'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY id ASC", (values,)) products = cur.fetchall()

# *Close Connection*

cur.close()

*if* request.method == 'POST' and form.validate():

name = form.name.data

mobile = form.mobile\_num.data order\_place = form.order\_place.data quantity = form.quantity.data

pid = request.args['order']

now = datetime.datetime.now() week = datetime.timedelta(days=7) delivery\_date = now + week

now\_time = delivery\_date.strftime("%y-%m-%d %H:%M:%S") # *Create Cursor*

curs = db.connection.cursor()

*if* 'uid' in session:

uid = session['uid']

curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s, %s)",

(uid, pid, name, mobile, order\_place, quantity, now\_time))

*else*:

curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s)",

(pid, name, mobile, order\_place, quantity, now\_time)) # *Commit cursor*

db.connection.commit() # *Close Connection*

cur.close()

# *sendemail()*

sendemail()

flash('Order successful', 'success')

*return* render\_template('wallet.html', wallet=products, form=form)

*if* 'view' in request.args:

q = request.args['view'] product\_id = q

x = content\_based\_filtering(product\_id) curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (q,)) products = curso.fetchall()

*return* render\_template('view\_product.html', x=x, tshirts=products)

*elif* 'order' in request.args: product\_id = request.args['order'] curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) product = curso.fetchall()

x = content\_based\_filtering(product\_id)

*return* render\_template('order\_product.html', x=x, tshirts=product, form=form)

*return* render\_template('wallet.html', wallet=products, form=form)

@*app*.*route*('/belt', methods=['GET', 'POST']) def *belt*():

form = OrderForm(request.form) # *Create cursor*

cur = db.connection.cursor() # *Get message*

values = 'belt'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY id ASC", (values,)) products = cur.fetchall()

# *Close Connection*

cur.close()

*if* request.method == 'POST' and form.validate(): name = form.name.data

mobile = form.mobile\_num.data order\_place = form.order\_place.data quantity = form.quantity.data

pid = request.args['order']

now = datetime.datetime.now() week = datetime.timedelta(days=7) delivery\_date = now + week

now\_time = delivery\_date.strftime("%y-%m-%d %H:%M:%S") # *Create Cursor*

curs = db.connection.cursor()

*if* 'uid' in session:

uid = session['uid']

curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s, %s)",

(uid, pid, name, mobile, order\_place, quantity, now\_time))

*else*:

curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s)",

(pid, name, mobile, order\_place, quantity, now\_time))

# *Commit cursor*

db.connection.commit()

# *Close Connection* cur.close() sendemail()

flash('Order successful', 'success')

*return* render\_template('belt.html', belt=products, form=form)

*if* 'view' in request.args:

q = request.args['view'] product\_id = q

x = content\_based\_filtering(product\_id) curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (q,)) products = curso.fetchall()

*return* render\_template('view\_product.html', x=x, tshirts=products)

*elif* 'order' in request.args: product\_id = request.args['order'] curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) product = curso.fetchall()

x = content\_based\_filtering(product\_id)

*return* render\_template('order\_product.html', x=x, tshirts=product, form=form)

*return* render\_template('belt.html', belt=products, form=form)

@*app*.*route*('/shoes', methods=['GET', 'POST']) def *shoes*():

form = OrderForm(request.form) # *Create cursor*

cur = db.connection.cursor() # *Get message*

values = 'shoes'

cur.execute("SELECT \* FROM products WHERE category=%s ORDER BY id ASC", (values,)) products = cur.fetchall()

# *Close Connection*

cur.close()

*if* request.method == 'POST' and form.validate(): name = form.name.data

mobile = form.mobile\_num.data order\_place = form.order\_place.data quantity = form.quantity.data

pid = request.args['order']

now = datetime.datetime.now() week = datetime.timedelta(days=7) delivery\_date = now + week

now\_time = delivery\_date.strftime("%y-%m-%d %H:%M:%S") # *Create Cursor*

curs = db.connection.cursor()

*if* 'uid' in session:

uid = session['uid']

curs.execute("INSERT INTO orders(uid, pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s, %s)",

(uid, pid, name, mobile, order\_place, quantity, now\_time))

*else*:

curs.execute("INSERT INTO orders(pid, ofname, mobile, oplace, quantity, ddate) " "VALUES(%s, %s, %s, %s, %s, %s)",

(pid, name, mobile, order\_place, quantity, now\_time)) # *Commit cursor*

db.connection.commit() # *Close Connection* cur.close()

sendemail()

flash('Order successful', 'success')

*return* render\_template('shoes.html', shoes=products, form=form)

*if* 'view' in request.args:

q = request.args['view'] product\_id = q

x = content\_based\_filtering(product\_id) curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (q,)) products = curso.fetchall()

*return* render\_template('view\_product.html', x=x, tshirts=products)

*elif* 'order' in request.args: product\_id = request.args['order'] curso = db.connection.cursor()

curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) product = curso.fetchall()

x = content\_based\_filtering(product\_id)

*return* render\_template('order\_product.html', x=x, tshirts=product, form=form)

*return* render\_template('shoes.html', shoes=products, form=form)

@*app*.*route*('/admin\_login', methods=['GET', 'POST']) @*not\_admin\_logged\_in*

def *admin\_login*():

*if* request.method == 'POST': # *GEt user form*

username = request.form['email'] password\_candidate = request.form['password']

# *Create cursor*

cur = db.connection.cursor() # *Get user by username*

result = cur.execute("SELECT \* FROM admin WHERE email=%s", [username])

*if* result > 0:

# *Get stored value*

data = cur.fetchone() password = data['password'] uid = data['id']

name = data['firstName']

# *Compare password if* (password):

# *passed* session['admin\_logged\_in'] = True session['admin\_uid'] = uid session['admin\_name'] = name

*return* redirect(url\_for('admin'))

*else*:

flash('Incorrect password', 'danger')

*return* render\_template('pages/login.html')

*else*:

flash('Username not found', 'danger') # *Close connection*

cur.close()

*return* render\_template('pages/login.html')

*return* render\_template('pages/login.html')

@*app*.*route*('/admin\_out') def *admin\_logout*():

*if* 'admin\_logged\_in' in session: session.clear()

*return* redirect(url\_for('admin\_login'))

*return* redirect(url\_for('admin'))

@*app*.*route*('/admin') @*is\_admin\_logged\_in* def *admin*():

curso = db.connection.cursor()

num\_rows = curso.execute("SELECT \* FROM products") result = curso.fetchall()

order\_rows = curso.execute("SELECT \* FROM orders")

users\_rows = curso.execute("SELECT \* FROM users")

*return* render\_template('pages/index.html', result=result, row=num\_rows, order\_rows=order\_rows, users\_rows=users\_rows)

@*app*.*route*('/orders') @*is\_admin\_logged\_in* def *orders*():

curso = db.connection.cursor()

num\_rows = curso.execute("SELECT \* FROM products") order\_rows = curso.execute("SELECT \* FROM orders") result = curso.fetchall()

users\_rows = curso.execute("SELECT \* FROM users")

*return* render\_template('pages/all\_orders.html', result=result, row=num\_rows, order\_rows=order\_rows, users\_rows=users\_rows)

@*app*.*route*('/users') @*is\_admin\_logged\_in* def *users*():

curso = db.connection.cursor()

num\_rows = curso.execute("SELECT \* FROM products") order\_rows = curso.execute("SELECT \* FROM orders") users\_rows = curso.execute("SELECT \* FROM users") result = curso.fetchall()

*return* render\_template('pages/all\_users.html', result=result, row=num\_rows, order\_rows=order\_rows, users\_rows=users\_rows)

@*app*.*route*('/admin\_add\_product', methods=['POST', 'GET']) @*is\_admin\_logged\_in*

def *admin\_add\_product*():

*if* request.method == 'POST': name = request.form.get('name') price = request.form['price']

description = request.form['description'] available = request.form['available'] category = request.form['category'] item = request.form['item']

code = request.form['code'] file = request.files['picture']

*if* name and price and description and available and category and item and code and file: pic = file.filename

photo = pic.replace("'", "") picture = photo.replace(" ", "\_")

*if* picture.lower().endswith(('.png', '.jpg', '.jpeg')): save\_photo = photos.save(file, folder=category) *if* save\_photo:

# *Create Cursor*

curs = db.connection.cursor() curs.execute("INSERT INTO

products(pName,price,description,available,category,item,pCode,picture)" "VALUES(%s, %s, %s, %s, %s, %s, %s, %s)",

(name, price, description, available, category, item, code, picture)) db.connection.commit()

product\_id = curs.lastrowid

curs.execute("INSERT INTO product\_level(product\_id)" "VALUES(%s)", [product\_id])

*if* category == 'tshirt':

level = request.form.getlist('tshirt')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format(field=lev)

curs.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'wallet':

level = request.form.getlist('wallet')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format(field=lev)

curs.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'belt':

level = request.form.getlist('belt')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format(field=lev)

curs.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'shoes':

level = request.form.getlist('shoes')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format(field=lev)

curs.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*else*:

flash('Product level not fund', 'danger')

*return* redirect(url\_for('admin\_add\_product')) # *Close Connection*

curs.close()

flash('Product added successful', 'success')

*return* redirect(url\_for('admin\_add\_product'))

*else*:

flash('Picture not save', 'danger')

*return* redirect(url\_for('admin\_add\_product'))

*else*:

flash('File not supported', 'danger')

*return* redirect(url\_for('admin\_add\_product'))

*else*:

flash('Please fill up all form', 'danger')

*return* redirect(url\_for('admin\_add\_product'))

*else*:

*return* render\_template('pages/add\_product.html')

@*app*.*route*('/edit\_product', methods=['POST', 'GET']) @*is\_admin\_logged\_in*

def *edit\_product*():

*if* 'id' in request.args:

product\_id = request.args['id'] curso = db.connection.cursor()

res = curso.execute("SELECT \* FROM products WHERE id=%s", (product\_id,)) product = curso.fetchall()

curso.execute("SELECT \* FROM product\_level WHERE product\_id=%s", (product\_id,)) product\_level = curso.fetchall()

*if* res:

*if* request.method == 'POST': name = request.form.get('name') price = request.form['price']

description = request.form['description'] available = request.form['available'] category = request.form['category'] item = request.form['item']

code = request.form['code'] file = request.files['picture'] # *Create Cursor*

*if* name and price and description and available and category and item and code and file: pic = file.filename

photo = pic.replace("'", "") picture = photo.replace(" ", "")

*if* picture.lower().endswith(('.png', '.jpg', '.jpeg')): file.filename = picture

save\_photo = photos.save(file, folder=category)

*if* save\_photo:

# *Create Cursor*

cur = db.connection.cursor() exe = curso.execute(

"UPDATE products SET pName=%s, price=%s, description=%s, available=%s, category=%s, item=%s, pCode=%s, picture=%s WHERE id=%s",

(name, price, description, available, category, item, code, picture, product\_id))

*if* exe:

*if* category == 'tshirt':

level = request.form.getlist('tshirt')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format( field=lev)

cur.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'wallet':

level = request.form.getlist('wallet')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format( field=lev)

cur.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'belt':

level = request.form.getlist('belt')

*for* lev *in* level: yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format( field=lev)

cur.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

*elif* category == 'shoes':

level = request.form.getlist('shoes')

*for* lev *in* level:

*else*:

yes = 'yes'

query = 'UPDATE product\_level SET {field}=%s WHERE product\_id=%s'.format( field=lev)

cur.execute(query, (yes, product\_id)) # *Commit cursor* db.connection.commit()

flash('Product level not fund', 'danger')

*return* redirect(url\_for('admin\_add\_product')) flash('Product updated', 'success')

*return* redirect(url\_for('edit\_product'))

*else*:

flash('Data updated', 'success')

*return* redirect(url\_for('edit\_product'))

*else*:

flash('Pic not upload', 'danger')

*return* render\_template('pages/edit\_product.html', product=product, product\_level=product\_level)

*else*:

flash('File not support', 'danger')

*return* render\_template('pages/edit\_product.html', product=product, product\_level=product\_level)

*else*:

flash('Fill all field', 'danger')

*return* render\_template('pages/edit\_product.html', product=product, product\_level=product\_level)

*else*:

*return* render\_template('pages/edit\_product.html', product=product, product\_level=product\_level)

*else*:

*return* redirect(url\_for('admin\_login'))

*else*:

*return* redirect(url\_for('admin\_login'))

@*app*.*route*('/search', methods=['POST', 'GET']) def *search*():

form = OrderForm(request.form)

*if* 'q' in request.args:

q = request.args['q'] # *Create cursor*

cur = db.connection.cursor() # *Get message*

query\_string = "SELECT \* FROM products WHERE pName LIKE %s ORDER BY id ASC" cur.execute(query\_string, ('%' + q + '%',))

products = cur.fetchall()

# *Close Connection*

cur.close()

flash('Showing result for: ' + q, 'success')

*return* render\_template('search.html', products=products, form=form)

*else*:

flash('Search again', 'danger')

*return* render\_template('search.html')

@*app*.*route*('/profile') @*is\_logged\_in*

def *profile*():

*if* 'user' in request.args:

q = request.args['user']

curso = db.connection.cursor()

curso.execute("SELECT \* FROM users WHERE id=%s", (q,)) result = curso.fetchone()

*if* result:

*if* result['id'] == session['uid']:

curso.execute("SELECT \* FROM orders WHERE uid=%s ORDER BY id ASC", (session['uid'],)) res = curso.fetchall()

*return* render\_template('profile.html', result=res)

*else*:

flash('Unauthorised', 'danger')

*return* redirect(url\_for('login'))

*else*:

flash('Unauthorised! Please login', 'danger')

*return* redirect(url\_for('login'))

*else*:

flash('Unauthorised', 'danger')

*return* redirect(url\_for('login'))

class UpdateRegisterForm(Form):

name = StringField('Full Name', [validators.length(min=3, max=50)], render\_kw={'autofocus': True, 'placeholder': 'Full Name'})

email = EmailField('Email', [validators.DataRequired(), validators.Email(), validators.length(min=4, max=25)],

render\_kw={'placeholder': 'Email'})

password = PasswordField('Password', [validators.length(min=3)], render\_kw={'placeholder': 'Password'})

mobile = StringField('Mobile', [validators.length(min=11, max=15)], render\_kw={'placeholder': 'Mobile'})

@*app*.*route*('/settings', methods=['POST', 'GET'])

@*is\_logged\_in* def *settings*():

form = UpdateRegisterForm(request.form)

*if* 'user' in request.args:

q = request.args['user']

curso = db.connection.cursor()

curso.execute("SELECT \* FROM users WHERE id=%s", (q,)) result = curso.fetchone()

*if* result:

*if* result['id'] == session['uid']:

*if* request.method == 'POST' and form.validate(): name = form.name.data

email = form.email.data

password = sha256\_crypt.encrypt(str(form.password.data)) mobile = form.mobile.data

# *Create Cursor*

cur = db.connection.cursor()

exe = cur.execute("UPDATE users SET name=%s, email=%s, password=%s, mobile=%s WHERE id=%s",

(name, email, password, mobile, q))

*if* exe:

flash('Profile updated', 'success')

*return* render\_template('user\_settings.html', result=result, form=form)

*else*:

flash('Profile not updated', 'danger')

*return* render\_template('user\_settings.html', result=result, form=form)

*else*:

flash('Unauthorised', 'danger')

*return* redirect(url\_for('login'))

*else*:

flash('Unauthorised! Please login', 'danger')

*return* redirect(url\_for('login'))

*else*:

flash('Unauthorised', 'danger')

*return* redirect(url\_for('login'))

class DeveloperForm(Form): #

id = StringField('', [validators.length(min=1)], render\_kw={'placeholder': 'Input a product id...'})

@*app*.*route*('/developer', methods=['POST', 'GET']) def *developer*():

form = DeveloperForm(request.form)

*if* request.method == 'POST' and form.validate(): q = form.id.data

curso = db.connection.cursor()

result = curso.execute("SELECT \* FROM products WHERE id=%s", (q,))

*if* result > 0:

x = content\_based\_filtering(q)

wrappered = wrappers(content\_based\_filtering, q) execution\_time = timeit.timeit(wrappered, number=0) seconds = ((execution\_time / 1000) % 60)

*return* render\_template('developer.html', form=form, x=x, execution\_time=seconds)

*else*:

nothing = 'Nothing found'

*return* render\_template('developer.html', form=form, nothing=nothing)

*else*:

*return* render\_template('developer.html', form=form)

*if* name == ' main ': app.run(debug=True)

# home.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<!-- */.col-lg-3* -->

<div *class*="col-lg-12">

<div *id*="carouselExampleIndicators" *class*="carousel slide my-4" *data-ride*="carousel">

<ol *class*="carousel-indicators">

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="0" *class*="active"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="1"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="2"></li>

</ol>

<div *class*="carousel-inner" *role*="listbox" *style*="max-height: 300px">

<div *class*="carousel-item active">

<img *class*="d-block img-fluid" *src*="/static/image/logo\_banner.jpg" *alt*="New York"

*width*="1100"

*height*="300"

*style*="object-fit:contain; background: #000; object-position: center; height: 300px;

width:1100px;">

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/bodyspray.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/t-shirt-wallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data- slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data- slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

<h3><a href="/tshirt">T-Shirt</a></h3>

<div class="row">

{% for product in tshirt %}

{% if product.category == 'tshirt' %}

<div class="col-lg-3 col-md-4 col-sm-6">

<div class="card mb-4">

<a href="/{{product.category}}?view={{product.id}}">

<div class="card-img-top id\_{{product.id}}"

style="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div class="card-body">

<h5><a class="card-title" href="javascript:void();">{{ product.pName}}</a></h5>

<h4>₹{{product.price}}</h4>

<span class="text-warning"></span>

</div>

</div>

<!-- /.card -->

</div>

{% endif %}

{% endfor %}

</div>

<!-- /.row -->

<div class="border p-4 mb-4" style="height: 350px">

<img class="w-100 h-100" src="/static/image/other/bodyspray.jpg" alt="New York">

</div>

<h3><a href="/wallet">Wallet</a></h3>

<div class="row">

{% for product in wallet %}

<div class="col-lg-3 col-md-4 col-sm-6">

<div class="card mb-4">

<a href="/{{product.category}}?view={{product.id}}">

<div class="card-img-top"

style="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div class="card-body">

<h5><a class="card-title" id="" href="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h5>

<h4>₹{{product.price}}</h4>

<span class="text-warning"></span>

</div>

</div>

<!-- /.card -->

</div>

{% endfor %}

</div>

<!-- /.row -->

<div class="border p-4 mb-4" style="height: 350px">

<img class="w-100 h-100" src="/static/image/other/watchwallpaper.jpg" alt="New York">

</div>

<h3><a href="/belt">Belt</a></h3>

<div class="row">

{% for product in belt %}

<div class="col-lg-3 col-md-4 col-sm-6">

<div class="card mb-4">

<a href="/{{product.category}}?view={{product.id}}">

<div class="card-img-top"

style="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div class="card-body">

<h5><a class="card-title" href="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h5>

<h4>₹{{product.price}}</h4>

<span class="text-warning"></span>

</div>

</div>

<!-- /.card -->

</div>

{% endfor %}

</div>

<div class="border p-4 mb-4" style="height: 350px">

<img class="w-100 h-100" src="/static/image/other/bodyspray.jpg" alt="New York">

</div>

<h3><a href="/shoes">Shoes</a></h3>

<div class="row">

{% for product in shoes %}

<div class="col-lg-3 col-md-4 col-sm-6">

<div class="card mb-4">

<a href="/{{product.category}}?view={{product.id}}">

<div class="card-img-top"

style="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div class="card-body">

<h5><a class="card-title" href="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h5>

<h4>₹{{product.price}}</h4>

<span class="text-warning"></span>

</div>

</div>

<!-- /.card -->

</div>

{% endfor %}

</div>

<div class="border p-4 mb-4" style="height: 350px">

<img class="w-100 h-100" src="/static/image/other/t-shirt-wallpaper.jpg" alt="New York">

</div>

</div>

<!-- /.col-lg-9 -->

</div>

<!-- /.row -->

</div>

<!-- /.container -->

{% endblock %}

# Admin\_login.html

{% extends 'layout.html' %}

{% block body %}

<div *class*="jumbotron">

<h2 *style*="text-align: center; color: #206aaa;">Login Form</h2>

{% from "includes/\_formhelpers.html" import render\_field %}

<div>

<form *method*="POST" *action*="">

<div *class*="form-group">

{{render\_field(form.username, class\_="form-control")}}

</div>

<div *class*="form-group">

<label><strong>Password</strong></label>

<input *type*="password" *name*="password" *class*="form-control" *value*={{request.form.password}}>

</div>

<button *type*="submit" *class*="btn btn-primary">Login</button>

</form>

</div>

</div>

{% endblock %}

**Belt.html**

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">Belt</h1>

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item">Wallet</a>

<a *href*="/belt" *class*="list-group-item active">Belt</a>

<a *href*="/shoes" *class*="list-group-item">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

{% include 'includes/\_flashmsg.html' %}

<div *id*="carouselExampleIndicators" *class*="carousel slide my-4" *data-ride*="carousel">

<ol *class*="carousel-indicators">

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="0" *class*="active"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="1"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="2"></li>

</ol>

<div *class*="carousel-inner" *role*="listbox" *style*="max-height: 300px">

<div *class*="carousel-item active">

<img *class*="d-block img-fluid" *src*="/static/image/other/watchwallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/bodyspray.jpg" *alt*="New York"

*width*="1100"

*height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/t-shirt-wallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

</div>

<a *class*="carousel-control-prev" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="prev">

<span *class*="carousel-control-prev-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Previous</span>

</a>

<a *class*="carousel-control-next" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="next">

<span *class*="carousel-control-next-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Next</span>

</a>

</div>

<div *class*="row">

{% for product in belt %}

<div *class*="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">

<div *class*="card h-100">

<a *href*="/belt?view={{product.id}}">

<div *class*="card-img-top"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>

</a>

<div *class*="card-body">

<h5 *class*="card-title">

<a *href*="/belt?view={{product.id}}">{{ product.pName}}</a>

</h5>

<h4>৳{{product.price}}</h4>

</div>

<div *class*="card-footer">

<small *class*="text-muted"></small>

</div>

</div>

</div>

{% endfor %}

</div>

<!-- */.row* -->

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# Layout.html

<html>

<head>

<meta *name*="viewport" *content*="width=device-width, initial-scale=1">

<title>Sundari Shopping</title>

<meta *charset*="utf-8">

<link *rel*="icon" *href*="{{ url\_for('static', filename='image/logo.png') }}" *type*="image/x-icon">

<meta *name*="viewport" *content*="width=device-width, initial-scale=1">

<link *rel*="stylesheet" *href*="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

<style>

body {

background: #60a7f8;

}

a {

color: rgb(65, 56, 56); text-decoration: none;

}

li {

color: white;

}

</style>

<script *src*="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script *src*="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

<script *src*="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>

<link *rel*="stylesheet" *href*="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body *style*=" margin-top: 57px ">

{% include 'includes/\_navbar.html' %}

<div *class*="container">

{% block body %} {% endblock %}

</div>

<script *src*="https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js"></script>

</body>

{% include 'includes/\_footer.html' %}

</html>

# Login.html

{% extends 'layout.html' %}

{% block body %}

{% include 'includes/\_flashmsg.html' %}

<div *class*="jumbotron">

<h2 *style*="text-align: center; color: #206aaa;">Login Form</h2>

{% from "includes/\_formhelpers.html" import render\_field %}

<div>

<form *method*="POST" *action*="">

<div *class*="form-group">

{{render\_field(form.username, class\_="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.password, class\_="form-control")}}

</div>

<button *type*="submit" *class*="btn btn-primary">Login</button>

</form>

</div>

</div>

{% endblock %}

**Orderproduct.html**

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3 my-4">

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item">Wallet</a>

<a *href*="/belt" *class*="list-group-item">Belt</a>

<a *href*="/shoes" *class*="list-group-item">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

<div *class*="row">

{% for product in tshirts %}

<div *class*="col-lg-6">

<div *class*="card mt-4">

<img *class*="card-img-top img-fluid"

*src*="static/image/product/{{product.category}}/{{product.picture}}" *alt*="">

</div>

</div>

<!-- */.card* -->

<div *class*="col-lg-6">

<div *class*="card card-outline-secondary my-4">

<div *class*="card-header"> Order Details

</div>

<div *class*="card-body">

{% from "includes/\_formhelpers.html" import render\_field %}

<form *method*="POST" *action*="/{{product.category}}?order={{product.id}}">

<div *class*="form-group">

{{render\_field(form.name, class\_="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.mobile\_num, class\_="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.quantity, class\_="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.order\_place, class\_="form-control")}}

</div>

<button *type*="submit" *class*="btn btn-primary">Confirm Order</button>

</form>

</div>

</div>

</div>

{% endfor %}

<!-- */.card* -->

</div>

{% endif %}

-->

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# Profile.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">{{session.s\_name}}</h1>

<div *class*="list-group">

<a *href*="/profile?user={{session.uid}}" *class*="list-group-item">Order List</a>

<a *href*="/settings?user={{session.uid}}" *class*="list-group-item">Settings</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

<div *class*="card card-default my-4">

<div *class*="card-header">

<i *class*="fa fa-bar-chart-o fa-fw"></i>My order list

</div>

<!-- */.panel-heading* -->

<div *class*="card-body">

<div *class*="row">

<div *class*="col-lg-12">

{% if result %}

<div *class*="table-responsive">

<table *class*="table table-bordered table-hover table-striped">

<thead>

<tr>

<th>Order No</th>

<th>Product ID</th>

<th>Quantity</th>

<th>Order Place</th>

<th>Mobile</th>

<th>Order Date</th>

<th>Delivery Date</th>

</tr>

</thead>

<tbody>

{% for order in result %}

<tr>

<td>{{order.id}}</td>

<td>{{order.pid}}</td>

<td>{{order.quantity}}</td>

<td>{{order.oplace}}</td>

<td>{{order.mobile}}</td>

<td>{{order.odate}}</td>

<td>{{order.ddate}}</td>

</tr>

{% endfor%}

</tbody>

</table>

</div>

{% else %}

<h3>No orders found</h3>

{% endif %}

<!-- */.table-responsive* -->

</div>

</div>

<!-- */.row* -->

</div>

<!-- */.panel-body* -->

</div>

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# Register.html

{% extends 'layout.html' %}

{% block body %}

<div *class*="jumbotron">

<h2 *style*="text-align: center; color: #206aaa;">Sign Up Form</h2>

{% from "includes/\_formhelpers.html" import render\_field %}

<form *method*="POST" *action*="">

<div *class*="form-group">

{{render\_field(form.name, class\_="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.email, class="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.username, class="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.password, class="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.mobile, class="form-control")}}

</div>

<p>

<input *type*="submit" *class*="btn btn-primary" *value*="Sign Up">

</p>

</form>

</div>

{% endblock %}

**Search.html**

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<!-- */.col-lg-3* -->

<div *class*="col-lg-12">

{% include 'includes/\_flashmsg.html' %}

<div *class*="row">

{% for product in products %}

<div *class*="col-lg-3 col-md-4 col-sm-6">

<div *class*="card mb-4">

<a *href*="/{{product.category}}?view={{product.id}}">

<div *class*="card-img-top"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div *class*="card-body">

<h3><a *class*="card-title" *id*="" *href*="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h3>

<h4>₹24.99</h4>

<span *class*="text-warning"></span>

</div>

</div>

<!-- */.card* -->

</div>

{% endfor %}

</div>

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# Shoes.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">Shoes</h1>

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item">Wallet</a>

<a *href*="/belt" *class*="list-group-item">Belt</a>

<a *href*="/shoes" *class*="list-group-item active">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

{% include 'includes/\_flashmsg.html' %}

<div *id*="carouselExampleIndicators" *class*="carousel slide my-4" *data-ride*="carousel">

<ol *class*="carousel-indicators">

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="0" *class*="active"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="1"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="2"></li>

</ol>

<div *class*="carousel-inner" *role*="listbox" *style*="max-height: 300px">

<div *class*="carousel-item active">

<img *class*="d-block img-fluid" *src*="/static/image/other/watchwallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/bodyspray.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/t-shirt-wallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

</div>

<a *class*="carousel-control-prev" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="prev">

<span *class*="carousel-control-prev-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Previous</span>

</a>

<a *class*="carousel-control-next" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="next">

<span *class*="carousel-control-next-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Next</span>

</a>

</div>

<div *class*="row">

{% for product in shoes %}

<div *class*="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">

<div *class*="card h-100">

<a *href*="/shoes?view={{product.id}}">

<div *class*="card-img-top"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div *class*="card-body">

<h5 *class*="card-title">

<a *href*="/shoes?view={{product.id}}">{{ product.pName}}</a>

</h5>

<h4>₹{{product.price}}</h4>

</div>

<div *class*="card-footer">

<small *class*="text-muted"></small>

</div>

</div>

</div>

{% endfor %}

</div>

<!-- */.row* -->

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# T-shirt.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">T-Shirt</h1>

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item active">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item">Wallet</a>

<a *href*="/belt" *class*="list-group-item">Belt</a>

<a *href*="/shoes" *class*="list-group-item">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

{% include 'includes/\_flashmsg.html' %}

<div *id*="carouselExampleIndicators" *class*="carousel slide my-4" *data-ride*="carousel">

<ol *class*="carousel-indicators">

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="0" *class*="active"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="1"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="2"></li>

</ol>

<div *class*="carousel-inner" *role*="listbox" *style*="max-height: 300px">

<div *class*="carousel-item active">

<img *class*="d-block img-fluid" *src*="/static/image/other/watchwallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/bodyspray.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/t-shirt-wallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

</div>

<a *class*="carousel-control-prev" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="prev">

<span *class*="carousel-control-prev-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Previous</span>

</a>

<a *class*="carousel-control-next" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="next">

<span *class*="carousel-control-next-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Next</span>

</a>

</div>

<div *class*="row">

{% for product in tshirt %}

<div *class*="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">

<div *class*="card h-100">

<a *href*="/tshirt?view={{product.id}}">

<div *class*="card-img-top"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div *class*="card-body">

<h5 *class*="card-title">

<a *href*="/tshirt?view={{product.id}}">{{ product.pName}}</a>

</h5>

<h4>₹{{product.price}}</h4>

</div>

<div *class*="card-footer">

<small *class*="text-muted"></small>

</div>

</div>

</div>

{% endfor %}

</div>

<!-- */.row* -->

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# User\_settings.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">{{session.s\_name}}</h1>

<div *class*="list-group">

<a *href*="/profile?user={{session.uid}}" *class*="list-group-item">Order List</a>

<a *href*="/settings?user={{session.uid}}" *class*="list-group-item">Settings</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

{% include 'includes/\_flashmsg.html' %}

<div *class*="card card-default my-4">

<div *class*="card-header">

<i *class*="fa fa-bar-chart-o fa-fw"></i>Update profile info

</div>

<!-- */.panel-heading* -->

<div *class*="card-body">

{% if result %}

{% from "includes/\_formhelpers.html" import render\_field %}

<form *method*="POST" *action*="/settings?user={{result.id}}">

<div *class*="form-group">

{{render\_field(form.name, class\_="form-control", value=result.name)}}

</div>

<div *class*="form-group">

{{render\_field(form.email, class="form-control", value=result.email)}}

</div>

<div *class*="form-group">

{{render\_field(form.password, class="form-control")}}

</div>

<div *class*="form-group">

{{render\_field(form.mobile, class="form-control", value=result.mobile)}}

</div>

<p>

<input *type*="submit" *class*="btn btn-primary" *value*="Update Settings">

</p>

</form>

{% endif%}

</div>

<!-- */.panel-body* -->

</div>

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# View\_product.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3 my-4">

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item">Wallet</a>

<a *href*="/belt" *class*="list-group-item">Belt</a>

<a *href*="/shoes" *class*="list-group-item">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

<div *class*="row">

{% for product in tshirts %}

<div *class*="col-lg-6">

<div *class*="card mt-4">

<img *class*="card-img-top img-fluid"

*src*="static/image/product/{{product.category}}/{{product.picture}}" *alt*="">

</div>

</div>

<!-- */.card* -->

<div *class*="col-lg-6">

<div *class*="card card-outline-secondary my-4">

<div *class*="card-header"> Product Details

</div>

<div *class*="card-body">

<h3 *class*="card-title">{{ product.pName}}</h3>

<h4>₹{{ product.price}}</h4>

<p *class*="card-text">{{ product.description}}</p>

<a *href*="/{{product.category}}?order={{product.id}}" *class*="btn btn-success">Order

Now</a>

</div>

</div>

</div>

{% endfor %}

<!-- */.card* -->

</div>

{% if x == '' %}

<h2 *class*="mb-4">No similar product found</h2>

{% else%}

<h2 *class*="mb-4">Similar product for you:</h2>

<div *class*="row">

{% for product in x[0] %}

<div *class*="col-lg-3 col-md-4 col-sm-6">

<div *class*="card mb-4">

<a *href*="/{{product.category}}?view={{product.id}}">

<div *class*="card-img-top id\_{{product.id}}"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div *class*="card-body">

<h5><a *class*="card-title" *href*="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h5>

<h4>₹{{product.price}}</h4>

</div>

</div>

<!-- */.card* -->

</div>

{% endfor %}

</div>

{% endif %}

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# Wallet.html

{% extends 'layout.html' %}

{% block body %}

<!-- *Page Content* -->

<div *class*="container">

<div *class*="row">

<div *class*="col-lg-3">

<h1 *class*="my-4">Wallet</h1>

<div *class*="list-group">

<a *href*="/tshirt" *class*="list-group-item">T-Shirt</a>

<a *href*="/wallet" *class*="list-group-item active">Wallet</a>

<a *href*="/belt" *class*="list-group-item">Belt</a>

<a *href*="/shoes" *class*="list-group-item ">Shoes</a>

</div>

</div>

<!-- */.col-lg-3* -->

<div *class*="col-lg-9">

{% include 'includes/\_flashmsg.html' %}

<div *id*="carouselExampleIndicators" *class*="carousel slide my-4" *data-ride*="carousel">

<ol *class*="carousel-indicators">

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="0" *class*="active"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="1"></li>

<li *data-target*="#carouselExampleIndicators" *data-slide-to*="2"></li>

</ol>

<div *class*="carousel-inner" *role*="listbox" *style*="max-height: 300px">

<div *class*="carousel-item active">

<img *class*="d-block img-fluid" *src*="/static/image/other/watchwallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/bodyspray.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

<div *class*="carousel-item">

<img *class*="d-block img-fluid" *src*="/static/image/other/t-shirt-wallpaper.jpg" *alt*="New York"

*width*="1100" *height*="300">

<div *class*="carousel-caption">

<h3>Los Angeles</h3>

<p>We had such a great time in LA!</p>

</div>

</div>

</div>

<a *class*="carousel-control-prev" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="prev">

<span *class*="carousel-control-prev-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Previous</span>

</a>

<a *class*="carousel-control-next" *href*="#carouselExampleIndicators" *role*="button" *data- slide*="next">

<span *class*="carousel-control-next-icon" *aria-hidden*="true"></span>

<span *class*="sr-only">Next</span>

</a>

</div>

<div *class*="row">

{% for product in wallet %}

<div *class*="col-12 col-sm-6 col-md-4 col-lg-4 col-xl-3 mb-4">

<div *class*="card h-100">

<a *href*="/wallet?view={{product.id}}">

<div *class*="card-img-top"

*style*="background: url('static/image/product/{{product.category}}/{{product.picture}}'); background-repeat:no-repeat; background-size: cover; height:200px;width:100% ">

</div>

</a>

<div *class*="card-body">

<h5 *class*="card-title">

<a *href*="/wallet?view={{product.id}}">{{ product.pName}}</a>

</h5>

<h4>₹{{product.price}}</h4>

</div>

<div *class*="card-footer">

<small *class*="text-muted"></small>

</div>

</div>

</div>

{% endfor %}

</div>

<!-- */.row* -->

</div>

<!-- */.col-lg-9* -->

</div>

<!-- */.row* -->

</div>

<!-- */.container* -->

{% endblock %}

# \_flashmsg.html

{% with messages = get\_flashed\_messages(with\_categories=true) %}

<!-- *Categories: success (green), info (blue), warning (yellow), danger (red)* -->

{% if messages %}

{% for category, message in messages %}

<div *class*="alert alert-{{ category }} alert-dismissible my-4" *role*="alert">

{{ message }}

</div>

{% endfor %}

{% endif %}

{% endwith %}

**\_footer.html**

<!-- *Footer* -->

<section *id*="footer">

<div *class*="container">

<div *class*="row text-center text-xs-center text-sm-left text-md-left">

<div *class*="col-xs-12 col-sm-4 col-md-4">

<h5>Quick Links</h5>

<ul *class*="list-unstyled quick-links">

<li><a *href*="/"><i *class*="fa fa-angle-double-right"></i>Home</a></li>

<li><a *href*="/tshirt"><i *class*="fa fa-angle-double-right"></i>T-Shirt</a></li>

<li><a *href*="/belt"><i *class*="fa fa-angle-double-right"></i>Belt</a></li>

<li><a *href*="/shoes"><i *class*="fa fa-angle-double-right"></i>Shoes</a></li>

<li><a *href*="/wallet"><i *class*="fa fa-angle-double-right"></i>Wallet</a></li>

</ul>

</div>

<div *class*="col-xs-12 col-sm-4 col-md-4">

<h5>Social</h5>

<ul *class*="list-unstyled quick-links">

<li><a *href*="https://[www.linkedin.com/in/vkparamesh3232](http://www.linkedin.com/in/vkparamesh3232)">Linked In</a></li>

<li><a *href*="https://github.com/Vkparamesh">Git Hub</a>

</li>

</ul>

</div>

<div *class*="col-xs-12 col-sm-4 col-md-4">

<h5>About Team</h5>

<ul *class*="list-unstyled quick-links">

<li>Team ID : PNT2022TMID38912</li>

<li>Team Leader : PARAMESHWARAN V K</li>

<li>Team member : MANOJ ADHITHYAN A</li>

<li>Team member : MOHANRAJ M</li>

<li>Team member : DAYANITHI S</li>

</ul>

</div>

</div>

<div *class*="row">

<div *class*="col-xs-12 col-sm-12 col-md-12 mt-2 mt-sm-2 text-center text-white">

<p *class*="h6"> krishnasamy college of engineering and technology &copy All right Reversed.</p>

</div>

</div>

</div>

</section>

<script>

*window.*watsonAssistantChatOptions = {

integrationID: "0319627f-d899-46cf-900d-e7d547432eff", // *The ID of this integration.*

region: "eu-gb", // *The region your integration is hosted in.*

serviceInstanceID: "2187bdc6-3463-4336-a0e5-105ac4d979db", // *The ID of your service instance. onLoad*: function (instance) { *instance.render*(); }

};

*setTimeout*(function () {

const *t* = *document.createElement*('script');

*t.*src = "https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (*window.watsonAssistantChatOptions.*clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

*document.head.appendChild*(t);

});

</script>

<!-- *./Footer* -->

# \_formhelper.html

{% macro render\_field(field) %}

<dt>{{ field.label }}</dt>

{{ field(\*\*kwargs)|safe }}

{% if field.errors %}

{% for error in field.errors %}

<span *class*="help-inline">{{ error }}</span>

{% endfor %}

{% endif %}

{% endmacro %}

**\_navbar.html**

<nav *class*="navbar navbar-expand-lg navbar-light bg-white border-bottom shadow-sm fixed-top">

<img *src*="/static/image/logo.png" *alt*="logo" *draggable*="false" *style*="width: 50px; padding-right:5px;">

<a *class*="navbar-brand" *href*="/">Sundari Shopping</a>

<button *class*="navbar-toggler" *type*="button" *data-toggle*="collapse" *data-target*="#navbarResponsive" *aria-controls*="navbarResponsive" *aria-expanded*="false" *aria-label*="Toggle navigation">

<span *class*="navbar-toggler-icon"></span>

</button>

<div *class*="collapse navbar-collapse" *id*="navbarResponsive">

<div *class*="input-group p-0 justify-content-center">

<form *action*="/search" *method*="get" *class*="w-100 m-0" *style*="padding-right: 20%;">

<div *class*="input-group-append rounded-left">

<input *type*="text" *class*="form-control rounded-left form-control-sm" *autocomplete*="off" *style*="float: left; border-radius: 0;" *name*="q" *placeholder*="What you want...">

<button *class*="btn btn-success btn-sm rounded-right" *style*="border-radius: 0;"

*type*="submit">Search</button>

</div>

</form>

</div>

<ul *class*="navbar-nav ml-auto">

{% if session.logged\_in %}

<li *class*="nav-item"><a *href*="/" *class*="nav-link"><span *class*=""></span> Home</a></li>

<li *class*="nav-item"><a *href*="/profile?user={{session.uid}}" *class*="nav- link">{{session.s\_name}}</a>

</li>

<li *class*="nav-item"><a *href*="/out" *class*="nav-link"><span *class*=""></span> Logout</a></li>

{% else %}

<li *class*="nav-item"><a *href*="/" *class*="nav-link"><span *class*=""></span> Home</a></li>

<li *class*="nav-item"><a *href*="/register" *class*="nav-link"><span *class*=""></span> Signup</a></li>

<li *class*="nav-item"><a *href*="/login" *class*="nav-link"><span *class*=""></span> Login</a></li>

{% endif %}

</ul>

</div>

</nav>

# Add\_product.html

{% extends 'pages/layout.html' %}

{% block body %}

<div *id*="page-wrapper">

<div *class*="row">

<div *class*="col-lg-12">

<h1 *class*="page-header">Add Product</h1>

{% include 'includes/\_flashmsg.html' %}

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-12">

<div *class*="panel panel-default">

<div *class*="panel-body">

<div *class*="row">

<div *class*="col-lg-6">

<form *role*="form" *enctype*=multipart/form-data *class*="" *action*="/admin\_add\_product" *method*="post">

<div *class*="form-group">

<label>Product Name</label>

<input *class*="form-control" *required*="required" *name*="name" *placeholder*="Enter

name">

price">

</div>

<div *class*="form-group">

<label>Price</label>

<input *class*="form-control" *required*="required" *name*="price" *placeholder*="Enter

</div>

<div *class*="form-group">

<label>Description</label>

<textarea *class*="form-control" *required*="required" *name*="description" *rows*="3" *placeholder*="Write product description"></textarea>

</div>

<div *class*="form-group">

<label>Available</label>

<input *class*="form-control" *required*="required" *name*="available" *placeholder*="Available">

</div>

<div *class*="form-group">

<label>Category</label>

<select *class*="form-control" *id*="category\_id" *onchange*="*populateList*();" *name*="category">

<option *value*="tshirt">T-Shirt</option>

<option *value*="wallet">Wallet</option>

<option *value*="belt">Belt</option>

<option *value*="shoes">Shoes</option>

</select>

</div>

<div *class*="form-group level-container">

<label>level for T-shirt</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="tshirt" *value*="v\_shape">V-Shape

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="tshirt" *value*="polo">Polo T-shirt

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="tshirt" *value*="clean\_text">Clean Text

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="tshirt" *value*="design">Colorful Design

</label>

</div>

<label>Level for Wallet</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="wallet" *value*="chain">Chain

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="wallet" *value*="leather">Leather

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="wallet" *value*="design">Design

</label>

</div>

<label>Level for Belt</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="belt" *value*="leather">Leather

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="belt" *value*="hook">Hook

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="belt" *value*="color">Color

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="belt" *value*="design">Design

</label>

</div>

<label>Level for Shoes</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="shoes" *value*="formal">Formal

</label>

<label *class*="checkbox-inline">

watch/shoes">

code">

<input *type*="checkbox" *name*="shoes" *value*="converse">Converse

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="shoes" *value*="loafer">Loafer

</label>

<label *class*="checkbox-inline">

<input *type*="checkbox" *name*="shoes" *value*="leather">Leather

</label>

</div>

</div>

<div *class*="form-group">

<label>Item</label>

<input *class*="form-control" *required*="required" *name*="item" *placeholder*="Ex:

</div>

<div *class*="form-group">

<label>Product Code</label>

<input *class*="form-control" *required*="required" *name*="code" *placeholder*="Enter

</div>

<div *class*="form-group">

<label>Add Image</label>

<input *type*="file" *required*="required" *name*="picture">

</div>

<button *type*="submit" *class*="btn btn-default">Submit Button</button>

</form>

</div>

<!-- */.col-lg-6 (nested)* -->

</div>

<!-- */.row (nested)* -->

</div>

<!-- */.panel-body* -->

</div>

<!-- */.panel* -->

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

</div>

<!-- */#page-wrapper* -->

{% endblock %}

# All\_orders.html

{% extends 'pages/layout.html' %}

{% block body %}

<div *id*="page-wrapper">

<div *class*="row">

<div *class*="col-lg-12">

<h1 *class*="page-header">Dashboard</h1>

{% include 'includes/\_flashmsg.html' %}

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-primary">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-list-ul fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{row}}</div>

<div>All Product</div>

</div>

</div>

</div>

<a *href*="/admin">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-green">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-tasks fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{users\_rows}}</div>

<div>All Users</div>

</div>

</div>

</div>

<a *href*="/users">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-yellow">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-shopping-cart fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{order\_rows}}</div>

<div>All Orders</div>

</div>

</div>

</div>

<a *href*="/orders">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-12">

<div *class*="panel panel-default my-4">

<div *class*="panel-heading">

<i *class*="fa fa-bar-chart-o fa-fw"></i>All Orders

</div>

<!-- */.panel-heading* -->

<div *class*="panel-body">

<div *class*="row">

<div *class*="col-lg-12">

{% if result %}

<div *class*="table-responsive">

<table *class*="table table-bordered table-hover table-striped">

<thead>

<tr>

<th>Id</th>

<th>User Id</th>

<th>Name</th>

<th>Product Id</th>

<th>Quantity</th>

<th>Order Place</th>

<th>Mobile</th>

<th>Order Date</th>

<th>Delivery Date</th>

</tr>

</thead>

<tbody>

{% for product in result %}

<tr>

<td>{{product.id}}</td>

<td>{{product.uid}}</td>

<td>{{product.ofname}}</td>

<td>{{product.pname}}</td>

<td>{{product.quantity}}</td>

<td>{{product.oplace}}</td>

<td>{{product.mobile}}</td>

<td>{{product.odate}}</td>

<td>{{product.ddate}}</td>

</tr>

{% endfor%}

</tbody>

</table>

</div>

{% else %}

<h3>No orders found</h3>

{% endif %}

<!-- */.table-responsive* -->

</div>

</div>

<!-- */.row* -->

</div>

<!-- */.panel-body* -->

</div>

</div>

</div>

<!-- */.row* -->

</div>

{% endblock %}

# All\_users.html

{% extends 'pages/layout.html' %}

{% block body %}

<div *id*="page-wrapper">

<div *class*="row">

<div *class*="col-lg-12">

<h1 *class*="page-header">Dashboard</h1>

{% include 'includes/\_flashmsg.html' %}

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-primary">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-list-ul fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{row}}</div>

<div>All Product</div>

</div>

</div>

</div>

<a *href*="/admin">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-green">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-tasks fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{users\_rows}}</div>

<div>All Users</div>

</div>

</div>

</div>

<a *href*="/users">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

<div *class*="col-lg-4 col-md-6">

<div *class*="panel panel-yellow">

<div *class*="panel-heading">

<div *class*="row">

<div *class*="col-xs-3">

<i *class*="fa fa-shopping-cart fa-5x"></i>

</div>

<div *class*="col-xs-9 text-right">

<div *class*="huge">{{order\_rows}}</div>

<div>All Orders</div>

</div>

</div>

</div>

<a *href*="/orders">

<div *class*="panel-footer">

<span *class*="pull-left">View Details</span>

<span *class*="pull-right"><i *class*="fa fa-arrow-circle-right"></i></span>

<div *class*="clearfix"></div>

</div>

</a>

</div>

</div>

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-12">

<div *class*="panel panel-default my-4">

<div *class*="panel-heading">

<i *class*="fa fa-bar-chart-o fa-fw"></i>All Orders

</div>

<!-- */.panel-heading* -->

<div *class*="panel-body">

<div *class*="row">

<div *class*="col-lg-12">

{% if result %}

<div *class*="table-responsive">

<table *class*="table table-bordered table-hover table-striped">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>Email</th>

<th>Username</th>

<th>Mobile</th>

<th>Registration Time</th>

<th>Online</th>

</tr>

</thead>

<tbody>

{% for product in result %}

<tr>

<td>{{product.id}}</td>

<td>{{product.name}}</td>

<td>{{product.email}}</td>

<td>{{product.username}}</td>

<td>{{product.mobile}}</td>

<td>{{product.reg\_time}}</td>

<td>{{product.online}}</td>

</tr>

{% endfor%}

</tbody>

</table>

</div>

{% else %}

<h3>No user found</h3>

{% endif %}

<!-- */.table-responsive* -->

</div>

</div>

<!-- */.row* -->

</div>

<!-- */.panel-body* -->

</div>

</div>

</div>

<!-- */.row* -->

</div>

{% endblock %}

# Edit\_product.html

{% extends 'pages/layout.html' %}

{% block body %}

<div *id*="page-wrapper">

<div *class*="row">

<div *class*="col-lg-12">

<h1 *class*="page-header">Edit Product</h1>

{% include 'includes/\_flashmsg.html' %}

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

<div *class*="row">

<div *class*="col-lg-12">

<div *class*="panel panel-default">

<div *class*="panel-body">

<div *class*="row">

<div *class*="col-lg-6">

{% for product in product%}

<form *role*="form" *enctype*=multipart/form-data *class*="" *action*="/edit\_product?id={{product.id}}"

*method*="post">

<div *class*="form-group">

<label>Product Name</label>

<input *class*="form-control" *required*="required" *name*="name" *value*="{{product.pName}}" *placeholder*="Enter name">

</div>

<div *class*="form-group">

<label>Price</label>

<input *class*="form-control" *required*="required" *name*="price" *value*="{{product.price}}" *placeholder*="Enter price">

</div>

<div *class*="form-group">

<label>Description</label>

<textarea *class*="form-control" *required*="required" *name*="description" *rows*="3" *placeholder*="Write product description">{{product.description}}</textarea>

</div>

<div *class*="form-group">

<label>Available</label>

<input *class*="form-control" *required*="required" *value*="{{product.available}}" *name*="available" *placeholder*="Available">

</div>

<div *class*="form-group">

<label>Category</label>

<select *class*="form-control" *id*="category\_id" *onchange*="*populateList*();" *name*="category">

{% if product.category == 'tshirt' %}

<option *value*="tshirt" *selected*>T-Shirt</option>

{% else %}

<option *value*="tshirt">T-Shirt</option>

{% endif %}

{% if product.category == 'wallet' %}

<option *value*="wallet" *selected*>Wallet</option>

{% else %}

<option *value*="wallet">Wallet</option>

{% endif %}

{% if product.category == 'belt' %}

<option *value*="belt" *selected*>Belt</option>

{% else %}

<option *value*="belt">Belt</option>

{% endif %}

{% if product.category == 'shoes' %}

<option *value*="shoes" *selected*>Shoes</option>

{% else %}

<option *value*="shoes">Shoes</option>

{% endif %}

</select>

</div>

{% for product\_level in product\_level %}

<div *class*="form-group level-container">

{% if product.category == 'tshirt' %}

<label>level for T-shirt</label>

<div *class*="form-group">

<label *class*="checkbox-inline active">

{% if product\_level.v\_shape == 'yes' %}

<input *type*="checkbox" *checked name*="tshirt" *value*="v\_shape">V-Shape

{% else%}

<input *type*="checkbox" *name*="tshirt" *value*="v\_shape">V-Shape

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.polo == 'yes' %}

<input *type*="checkbox" *checked name*="tshirt" *value*="polo">Polo T-shirt

{% else%}

<input *type*="checkbox" *name*="tshirt" *value*="polo">Polo T-shirt

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.clean\_text == 'yes' %}

<input *type*="checkbox" *checked name*="tshirt" *value*="clean\_text">Clean Text

{% else%}

<input *type*="checkbox" *name*="tshirt" *value*="clean\_text">Clean Text

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.design == 'yes' %}

<input *type*="checkbox" *checked name*="tshirt" *value*="design">Colorful Design

{% else%}

<input *type*="checkbox" *name*="tshirt" *value*="design">Colorful Design

{% endif %}

</label>

</div>

{% endif %}

{% if product.category == 'wallet' %}

<label>Level for Wallet</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

{% if product\_level.chain == 'yes' %}

<input *type*="checkbox" *checked name*="wallet" *value*="chain">Chain

{% else%}

<input *type*="checkbox" *name*="wallet" *value*="chain">Chain

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.leather == 'yes' %}

<input *type*="checkbox" *checked name*="wallet" *value*="leather">Leather

{% else%}

<input *type*="checkbox" *name*="wallet" *value*="leather">Leather

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.design == 'yes' %}

<input *type*="checkbox" *checked name*="wallet" *value*="design">Design

{% else%}

<input *type*="checkbox" *name*="wallet" *value*="design">Design

{% endif %}

</label>

</div>

{% endif %}

{% if product.category == 'belt' %}

<label>Level for Belt</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

{% if product\_level.leather == 'yes' %}

<input *type*="checkbox" *checked name*="belt" *value*="leather">Leather

{% else%}

<input *type*="checkbox" *name*="belt" *value*="leather">Leather

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.hook == 'yes' %}

<input *type*="checkbox" *checked name*="belt" *value*="hook">Hook

{% else%}

<input *type*="checkbox" *name*="belt" *value*="hook">Hook

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.color == 'yes' %}

<input *type*="checkbox" *checked name*="belt" *value*="color">Color

{% else%}

<input *type*="checkbox" *name*="belt" *value*="color">Color

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.design == 'yes' %}

<input *type*="checkbox" *checked name*="belt" *value*="design">Design

{% else%}

<input *type*="checkbox" *name*="belt" *value*="design">Design

{% endif %}

</label>

</div>

{% endif %}

{% if product.category == 'shoes' %}

<label>Level for Shoes</label>

<div *class*="form-group">

<label *class*="checkbox-inline">

{% if product\_level.formal == 'yes' %}

<input *type*="checkbox" *checked name*="shoes" *value*="formal">Formal

{% else%}

<input *type*="checkbox" *name*="shoes" *value*="formal">Formal

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.converse == 'yes' %}

<input *type*="checkbox" *checked name*="shoes" *value*="converse">Converse

{% else%}

<input *type*="checkbox" *name*="shoes" *value*="converse">Converse

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.loafer == 'yes' %}

<input *type*="checkbox" *checked name*="shoes" *value*="loafer">Loafer

{% else%}

<input *type*="checkbox" *name*="shoes" *value*="loafer">Loafer

{% endif %}

</label>

<label *class*="checkbox-inline">

{% if product\_level.leather == 'yes' %}

<input *type*="checkbox" *name*="shoes" *checked value*="leather">Leather

{% else%}

<input *type*="checkbox" *name*="shoes" *value*="leather">Leather

{% endif %}

</label>

</div>

{% endif %}

</div>

{% endfor %}

<div *class*="form-group">

<label>Item</label>

<input *class*="form-control" *required*="required" *name*="item" *value*="{{product.item}}" *placeholder*="Ex: watch/shoes">

</div>

<div *class*="form-group">

<label>Product Code</label>

<input *class*="form-control" *required*="required" *value*="{{product.pCode}}" *name*="code" *placeholder*="Enter code">

</div>

<div *class*="form-group">

<label>Add Image</label>

<input *type*="file" *name*="picture">

</div>

<button *type*="submit" *class*="btn btn-default">Submit Button</button>

</form>

{% endfor %}

</div>

<!-- */.col-lg-6 (nested)* -->

</div>

<!-- */.row (nested)* -->

</div>

<!-- */.panel-body* -->

</div>

<!-- */.panel* -->

</div>

<!-- */.col-lg-12* -->

</div>

<!-- */.row* -->

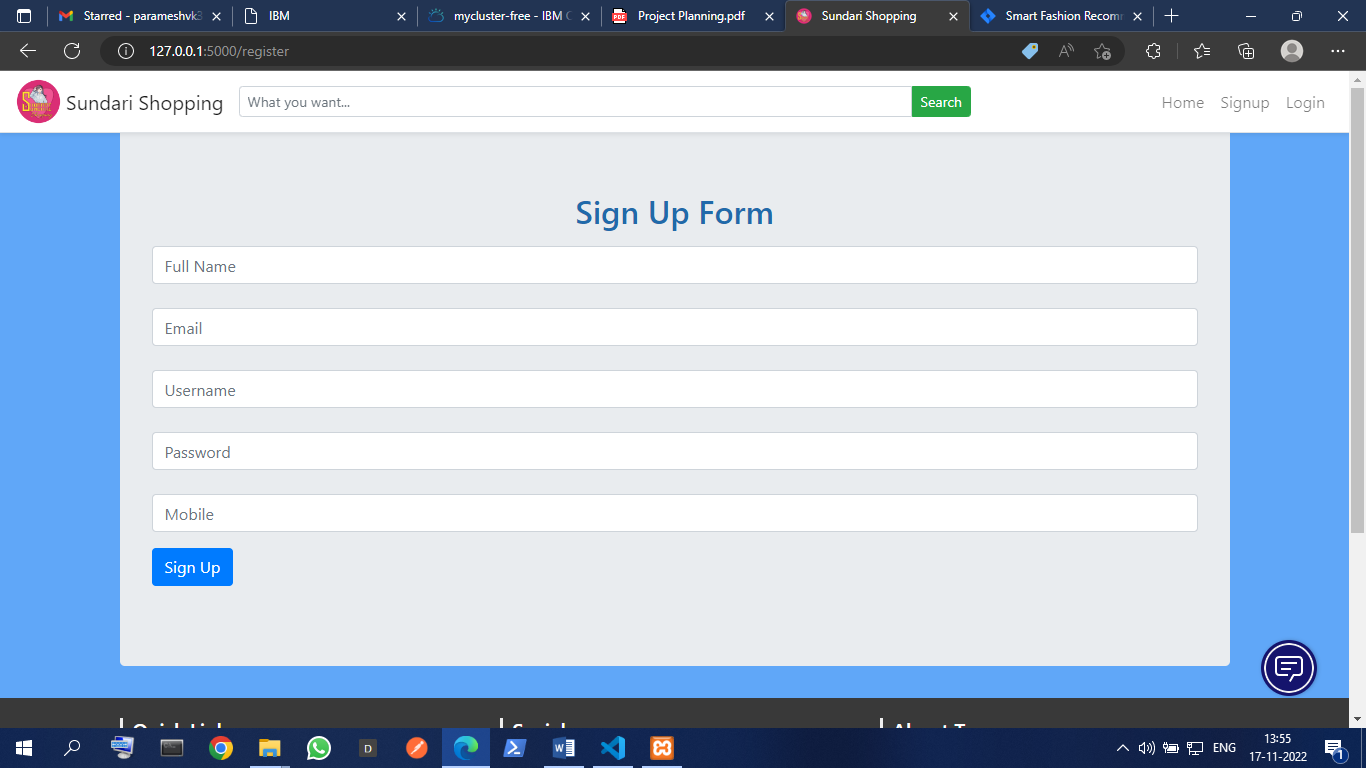
</div>

<!-- */#page-wrapper* -->

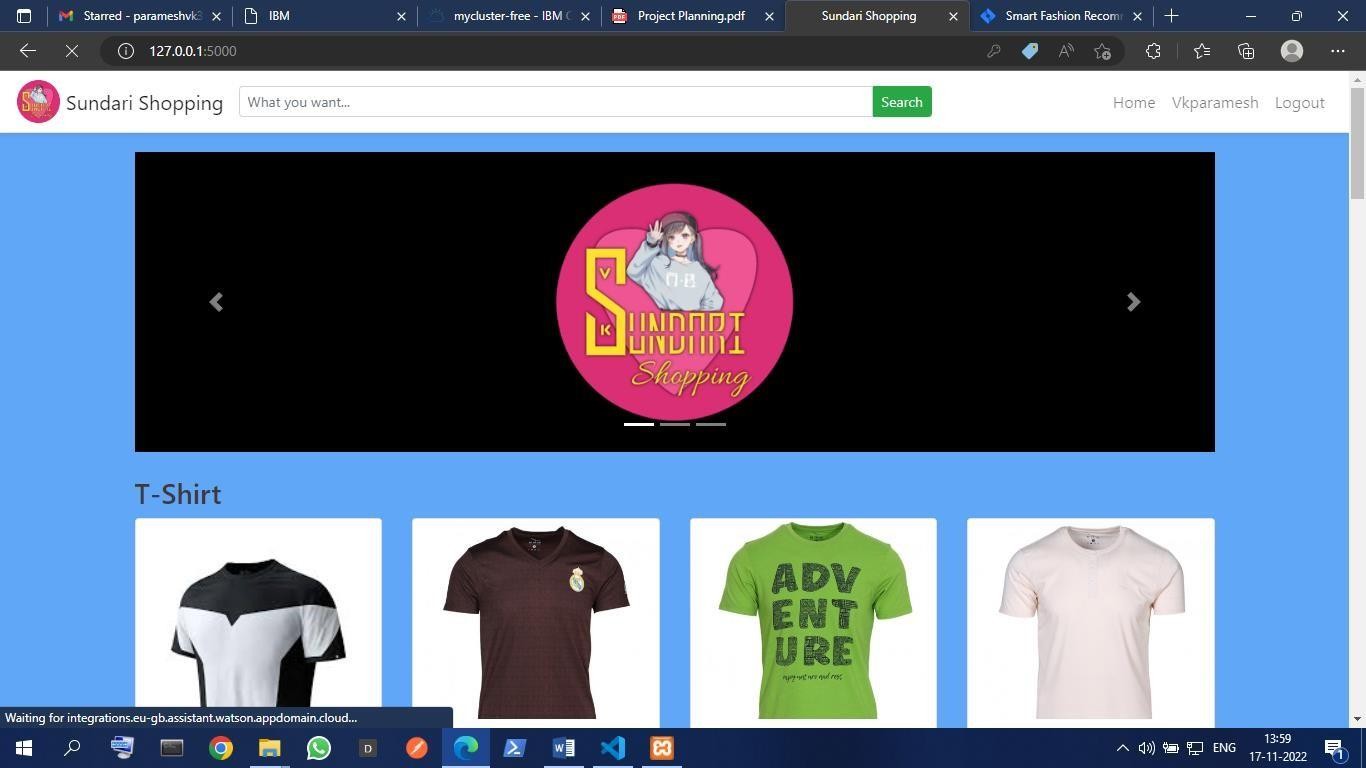
{% endblock %}

# SCREENSHOTS

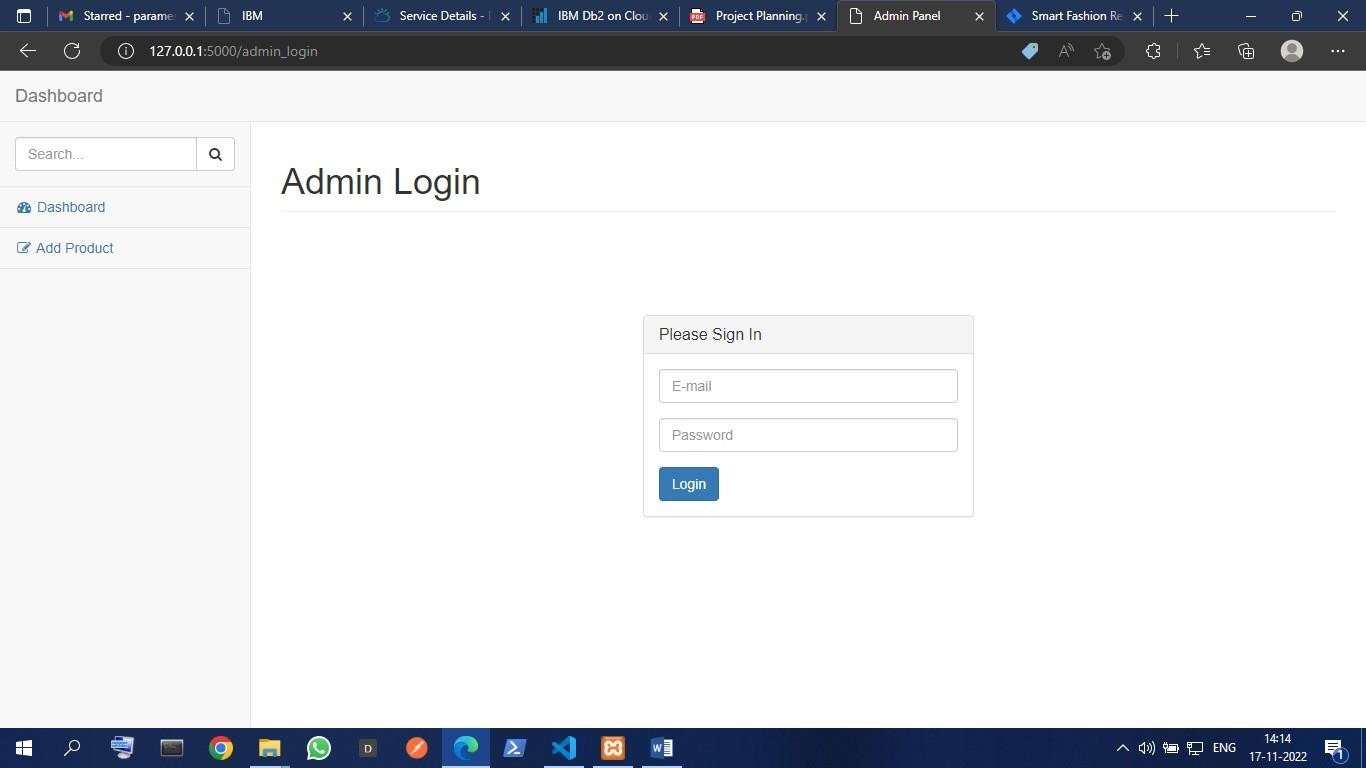
**Register page**



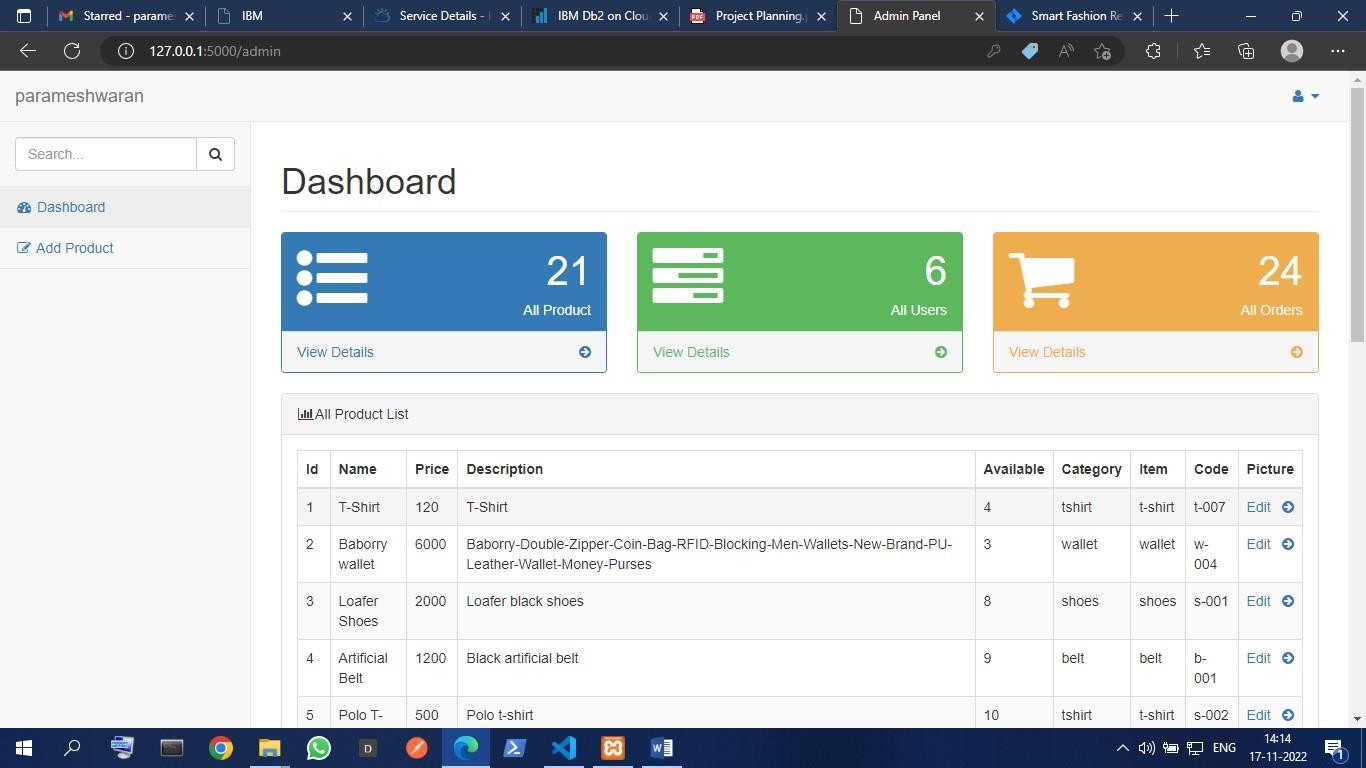
# Home page



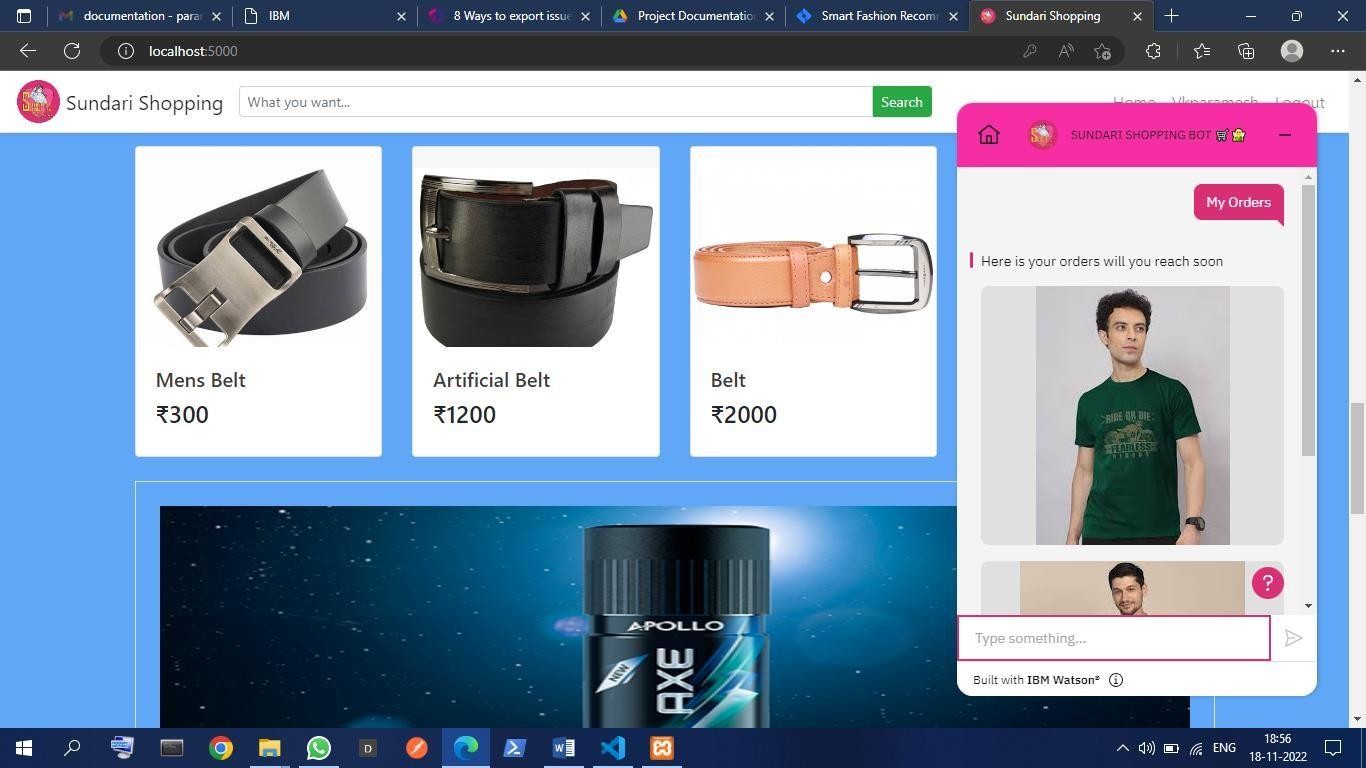
**Admin Login Page**

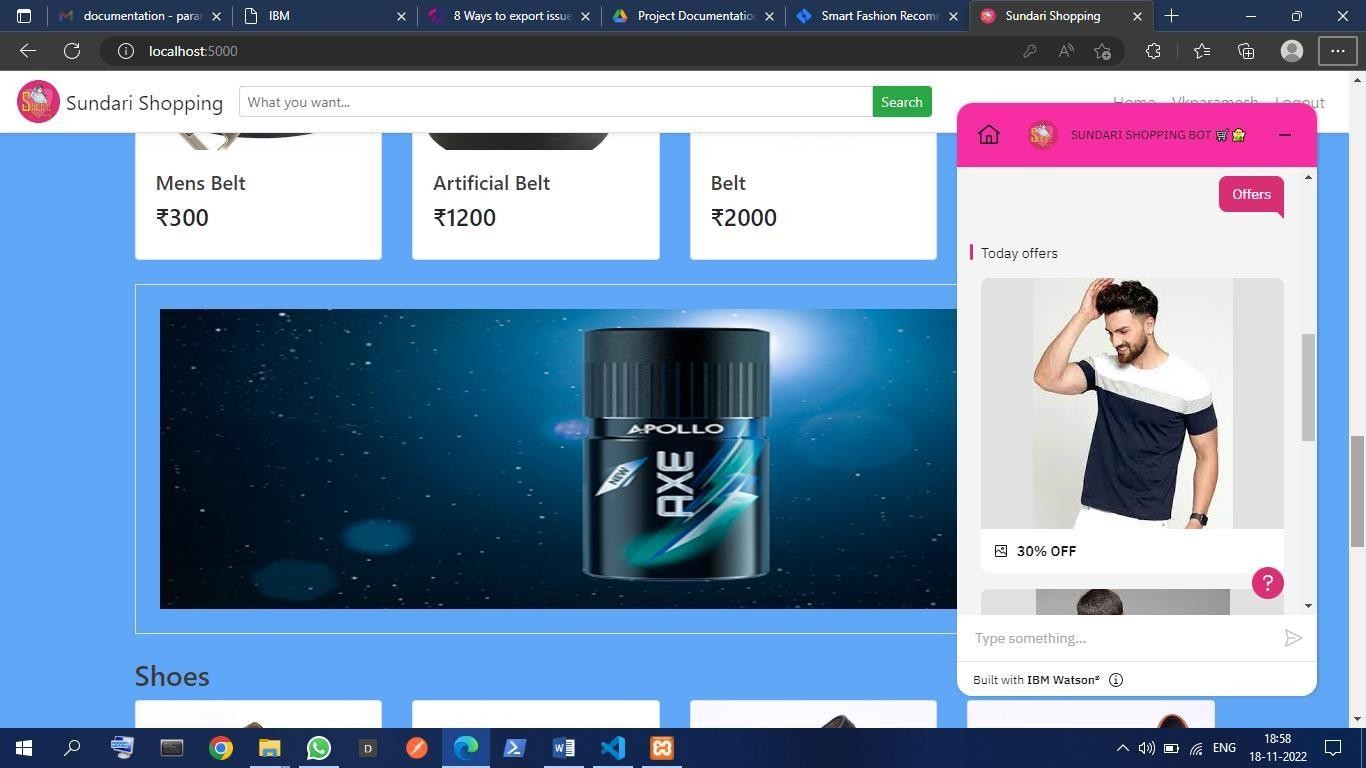


# Admin Dashboard Page



**Integrate Chat Bot**





# GitHub & Project Demo Link

* + Our GitHub Repository Direct Link
  + <https://github.com/IBM-EPBL/IBM-Project-37487-1660310518>
  + Project Demonstration Video Direct Link

## shorturl.at/eiGM4