Implementing Web Application

Team ID	PNT2022TMID22126	
Project Name	Smart Fashion Recommender	
	Application	

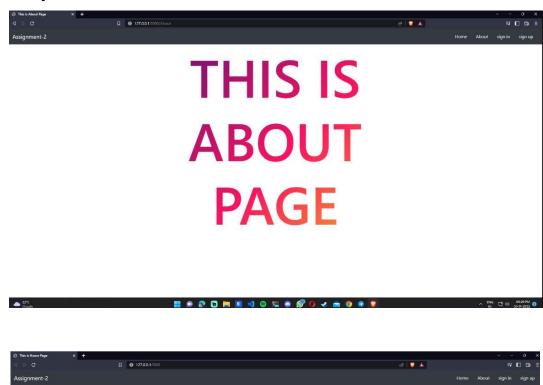
Create UI To Interact With Application:

```
from flask import Flask, render template from
flask_navigation import Navigation
from flask import Flask, render template, request, redirect, url for,
session
import
ibm_db import
bcrypt conn =
ibm db.connect("DATABASE=bludb;HOSTNAME=;PORT=;SECURITY=SSL;SSLServerCertificate=
DigiCertGlobalRootCA.crt;UID=;PWD=",'','')
# url_for('static', filename='style.css')
app = Flask(_name_) app.secret_key =
b' 5#y2L"F4Q8z\n\xec]/'
nav = Navigation(app)
nav.Bar('top', [
nav.Item('Home', 'index'),
nav.Item('About','about'),
nav.Item('Sign in','signin'),
nav.Item('Sign up','signup'),
])
@app.route('/') def
index():
  return render_template('index.html')
@app.route('/navpage') def
navpage():
    return render_template('navpage.html')
```

```
@app.route('/About') def about():
  return render_template('About.html')
@app.route('/Sign up') def signup():
 @app.route("/",methods=['GET']) def home():
if 'email' not in session:
     return redirect(url_for('login'))
                                         return
render_template('home.html',name='Home')
@app.route("/register",methods=['GET','POST']) def
register(): if request.method == 'POST':
                                         email =
request.form['email']
                        username =
request.form['username']
                           rollNo =
request.form['rollNo']
                         password =
request.form['password']
    if not email or not username or not rollNo or not password:
     return render template('register.html',error='Please fill all fields')
        hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM USER WHERE email=? OR rollNo=?"
                                                            stmt =
ibm_db.prepare(conn, query)
                             ibm_db.bind_param(stmt,1,email)
ibm db.bind param(stmt,2,rollNo)
                                  ibm db.execute(stmt)
                                                          isUser =
ibm_db.fetch_assoc(stmt)
        if not isUser:
     insert_sql = "INSERT INTO User(username,email,PASSWORD,rollNo) VALUES
           prep_stmt = ibm_db.prepare(conn, insert sql)
ibm db.bind_param(prep_stmt, 1, username)
                                            ibm_db.bind_param(prep_stmt, 2,
          ibm_db.bind_param(prep_stmt, 3, hash)
return render_template('register.html', success="You can login")
```

```
return render template('register.html',error='Invalid Credentials')
   return render_template('register.html',name='Home')
@app.route("/login", methods=['GET', 'POST']) def
login():
            if request.method == 'POST':
                                               email
= request.form['email']
                             password =
request.form['password']
      if not email or not password:
        return render_template('login.html',error='Please fill all fields')
query = "SELECT * FROM USER WHERE email=?" stmt = ibm db.prepare(conn, query)
ibm db.bind param(stmt,1,email)
                                     ibm db.execute(stmt)
                                                               isUser =
ibm db.fetch assoc(stmt)
                          print(isUser,password)
      if not isUser:
        return render_template('login.html',error='Invalid Credentials')
             isPasswordMatch =
bcrypt.checkpw(password.encode('utf8'),isUser['PASSWORD'].encode('utf-
8'))
       if not isPasswordMatch:
        return render template('login.html',error='Invalid Credentials')
       session['email'] = isUser['EMAIL']
return redirect(url_for('home'))
     return render template('login.html',name='Home')
@app.route('/logout') def logout():
    session.pop('email', None) return
redirect(url_for('login'))
  return render template('signup.html')
if _name_ == ' main_':
  app.run()
```

Outputs:



THIS IS HOME PAGE



