JOB AND SKILL RECOMMENDER APPLICATION

Date	07 November 2022
Team ID	PNT2022TMID31580
Project Name	Job & skill Recommender Application

TEAM MEMBERS:

Team Lead : Dharunraj S

Team Member 1: Aakash Veeraiah G

Team Member 2: Abishek R

Team Member 3: Arunaprakash K S

CONTENTS

1. INTRODUCTION

Nowadays, most people do their job searches on the Internet, using sites such as Linkedin1, Indeed2, and others. Typically, a job

A job seeker can use these sites in two ways:

1) Conducting a search based on keywords connected to the job position that he or she is seeking
2) Creating and/or updating a professional profile comprising information on his/her education, professional experience, professional abilities, and other qualifications, and receiving customized employment recommendations based on this information. Sites that support the former case are more popular and have a simpler structure; however, their suggestions are less accurate than those of sites that use profile data.
3) Personalized job suggestion sites used a range of recommender systems, including content-based filtering, collaborative filtering, knowledge-based, and hybrid techniques.

.

1.1 Project Overview

It is quite tough to get a job based on one's skills because the employment market is extremely competitive. Furthermore, some people are unaware of accessible job posts in which they can employ their expertise. A skill and job recommender are a web application that provides users with job posting recommendations based on their skills.

1.2 Purpose

- Better User Experience
- Higher Engagement
- Skill Improvement
- Revenue Opportunities
- Job Opportunities

2. LITERATURE SURVEY

2.1 Existing Problem

The created system is made up of three modules: a college campus recruitment system, a keyword-based search from online recruitment sites, and an Android application. Student and employer profiles are collected in the college campus recruitment system. Students' profiles are created by collecting information from students during the registration and login process. The admin will create the company profile based on the information and requirements provided by the organization. Following that, profile matching is performed on the profiles of the students and the company. Students can look for companies from numerous online recruitment sites in the second module, which is a keyword-based search module. These sites are searched using the web crawling approach. Students must enter a keyword, such as C#, and a web crawler will look for organizations that have that keyword. Students must enter a keyword, such as C#, and the web crawler will look for organizations that have openings for C# developers on various online recruitment sites such as Naukri.com.

2.2 References

S. No	Title	Publication details	Methodology/ algorithms	Merits	Demerits
1.	Shaha T Al-Otaibi and Mourad Ykhlef. 2012. A survey of job recommender systems.	International Journal of Physical Sciences, Vol. 7, 29 (2012)	The recommender systems used to determine the interested items for a specific user by employing a variety of information resources that is related to users and items.	Can provide accurate results based on user's interests.	The number of recommendations is limited.
2.	A Life-long Learning Recommender System to Promote Employability	iJET – Vol. 12, No. 6, 2017 77	Provide analytical tools that support people to be aware of how well positioned they are for succeeding in their professional and job expectations	Usage of cloud server data base Low power consumption Microcontroller and user friendly	Lack of chatbot system Difficult to analyse skill levels of the user
3.	Dynamic User Profile-Based Job Recommender System	Wenxing Hong, Siting Zheng, Huan Wang School of Information Science and Technology Xiamen University Xiamen, China (ICCSE 2013) April 26-28, 2013.	The basic features are extracted from the job seeker's profile. The profile might get out-dated when the user does not update it in a timely manner. So, the system makes a statistic at regular intervals, to generate the dynamic basic features.	Job applicants do not update their profile in a timely manner. This system aims at updating and extending the user profile dynamically based on the historical applied jobs and the behaviour of job applicants.	Besides the time and the dimensionality of features, there are other factors that affect the dynamic job recommendation system.

4.	Collaborative Job Prediction based on Naive Bayes Classifier using Python Platform	Choudhary, S., Koul, S., Mishra, S., Thakur, A., & Jain, R. (2016, October). Collaborative job prediction based on Naïve Bayes Classifier using python platform.	The system is designed to suggest the jobs to the user depending upon his profile and by calculating a similarity index using Euclidean distance of two skill sets and then ranking them according to their naïve Bayes algorithm	It has small computational overhead compared to Machine learning models	Susceptible to cold-start problem
5.	Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale	Shalaby, W., AlAila, B., Korayem, M., Pournajaf, L., AlJadda, K., Quinn, S., & Zadrozny, W. (2017, December). In 2017 IEEE international conference on big data (big data) (pp. 1544-1553). IEEE.	The proposed approach incorporates content-based signals to the CF-based core system (hybrid recommendation).	This approach overcomes the major challenges of sparsity and scalability. It overcomes the problem of cold start by harnessing the power of deep learning in addition to user behaviour to serve hybrid recommendations.	Many of the parameters in the system were decided heuristically and with manual evaluation.
6.	Job Recommendation through Progression of Job Selection	A. Nigam, A. Roy, H. Singh and H. Waila, "Job Recommendation through Progression of Job Selection	The framework consisted of the website module, brain module and worker module. The brain composes the recommendations using a blended approach of machine learning and non-machine learning strategies.	BLSTM-A surpassed the other models in terms of F1 score. It overcomes the cold start problem by using fuzzy matching	Less accuracy

7.	A Combined Representation Learning Approach for Better Job and Skill Recommendation	Dave, V. S., Zhang, B., Al Hasan, M., AlJadda, K., & Korayem, M. (2018, October).	The proposed solution leverages the information of three graphs in order to represent each job and skill in a shared low-dimensional vector space for solving the job recommendation task from the historical job data. .	The proposed embedding methodology consistently outperforms three state-of-the art methods in terms of job recommendation task .	The proposed representation learning framework is transudative
8.	Job Recommendation based on Job Seeker Skills: An Empirical Study	Jorge Valverde-Rebaza, Ricardo Puma, Paul Bustios, Nathalia C Silva. Conference: March 2018-	Job recommendation is performed using TF-IDF and four different configurations of Word2Vec over a dataset of job seeker profiles and job vacancies	Personalized job recommendation is done based on the job seeker's profile Recommendations based on other data like query based on keywords related to the job vacancy	TF-IDF computes document similarity directly in word-count space. It makes no use of semantic similarities between words.
9.	User click prediction for personalized job recommendation	Miao Jiang, Yi Fang, Huangming	Predicting the probability that a given user clicks on jobs is crucial to job search engines as the predictions can be used to provide personalized job recommendations for job seekers	An accurate machine learning based click model relies on the careful design of features.	Lack of features such as dwell time, query reformulation, and the sequence of clicks

2.3 Problem Statement Definition

The problem statement specifies the current state, the anticipated future state, and any gaps in between. A problem statement is an important communication tool that can help guarantee that everyone working on a project understands the problem and why the project is important.

3. IDEATION AND PROPOSED SOLUTION

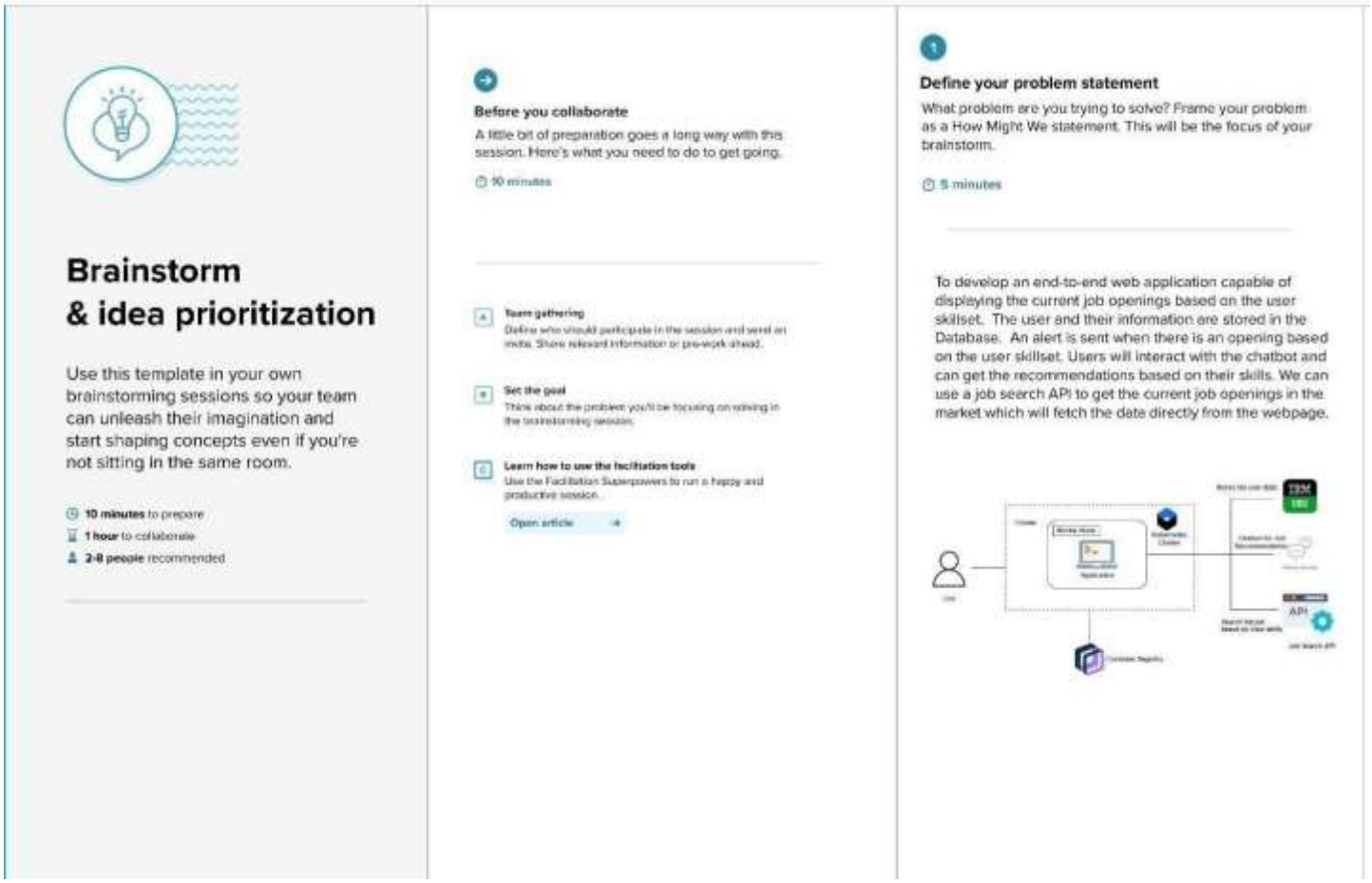
3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

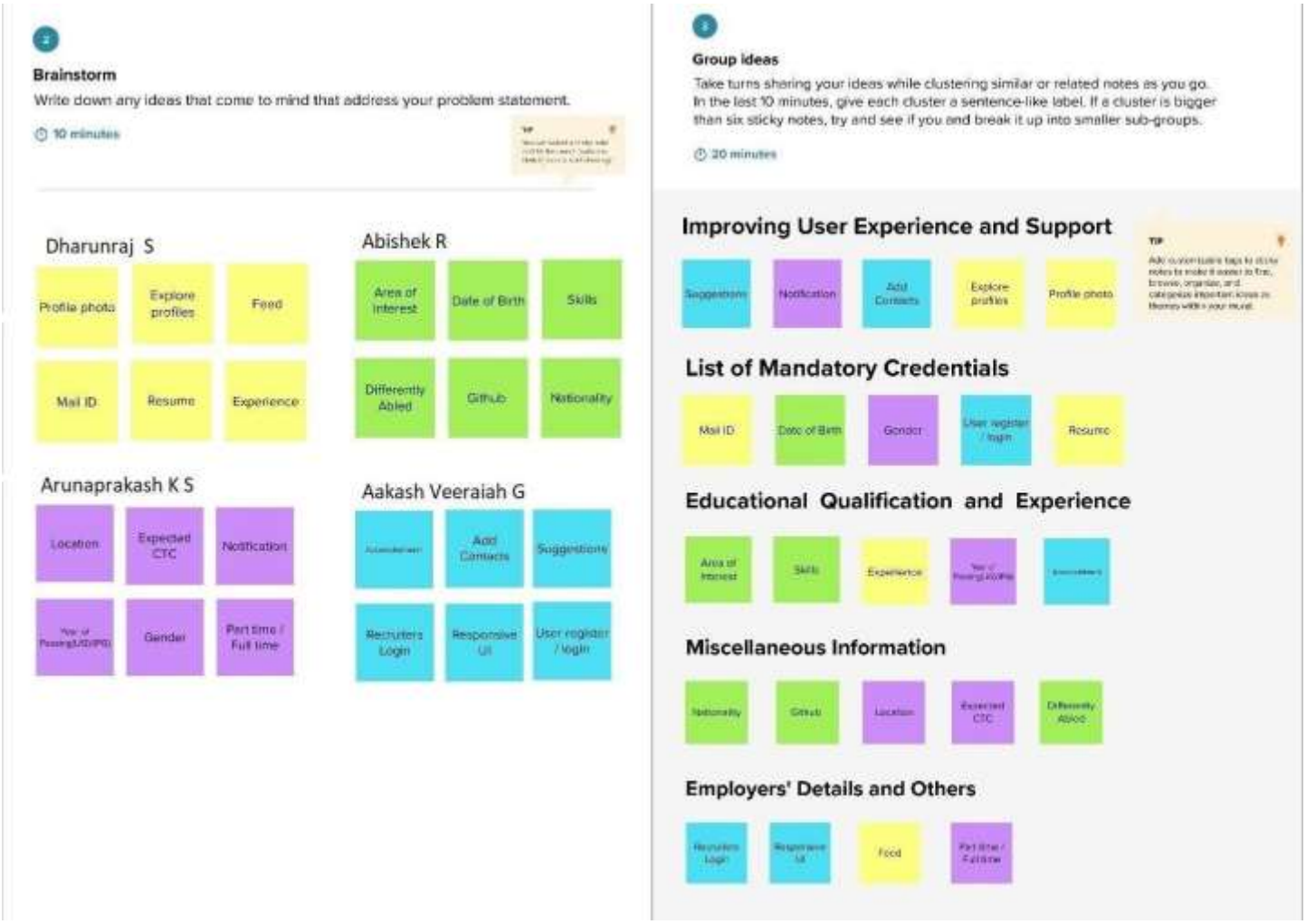
3.2 Ideation And Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



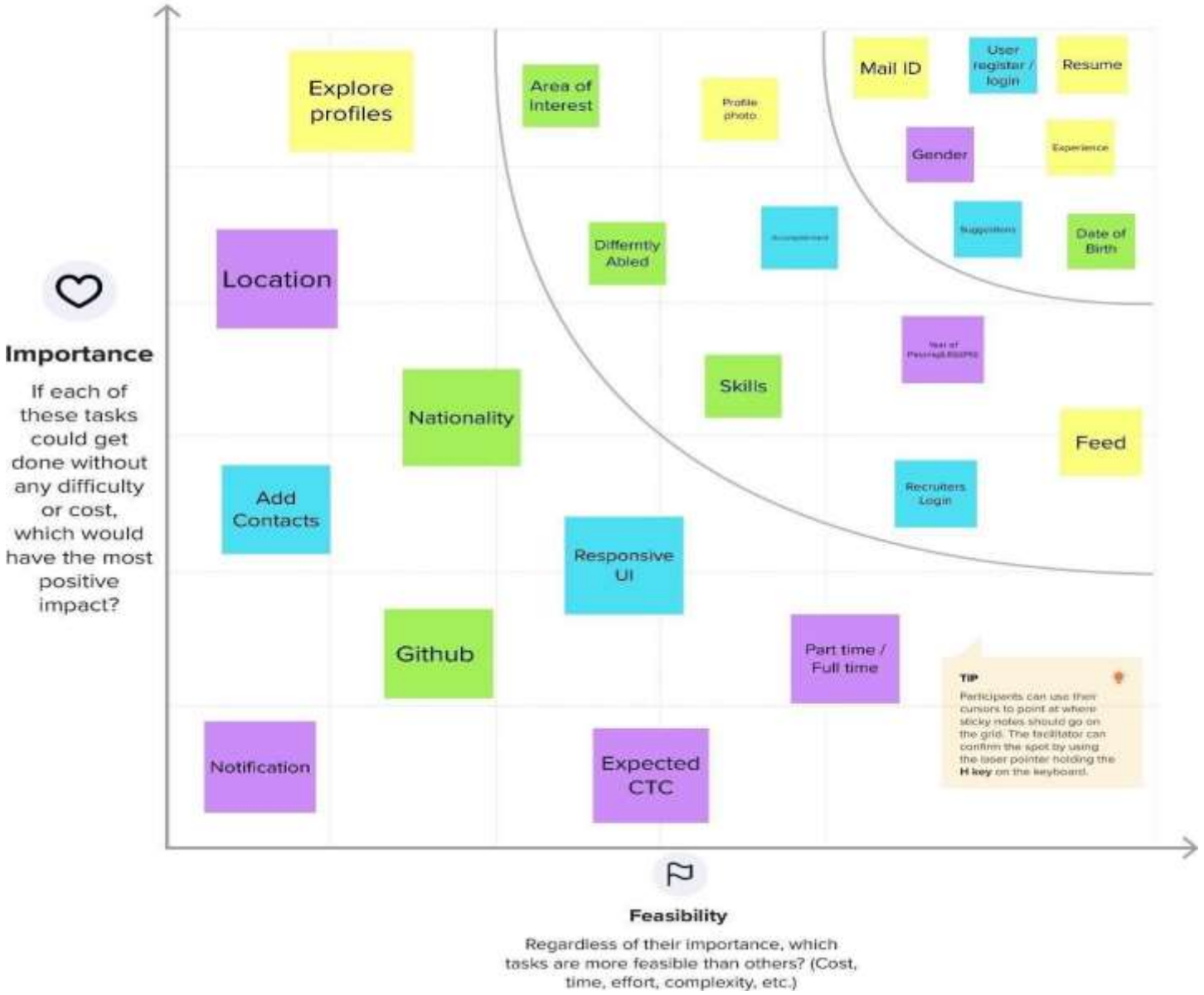
Step-3: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

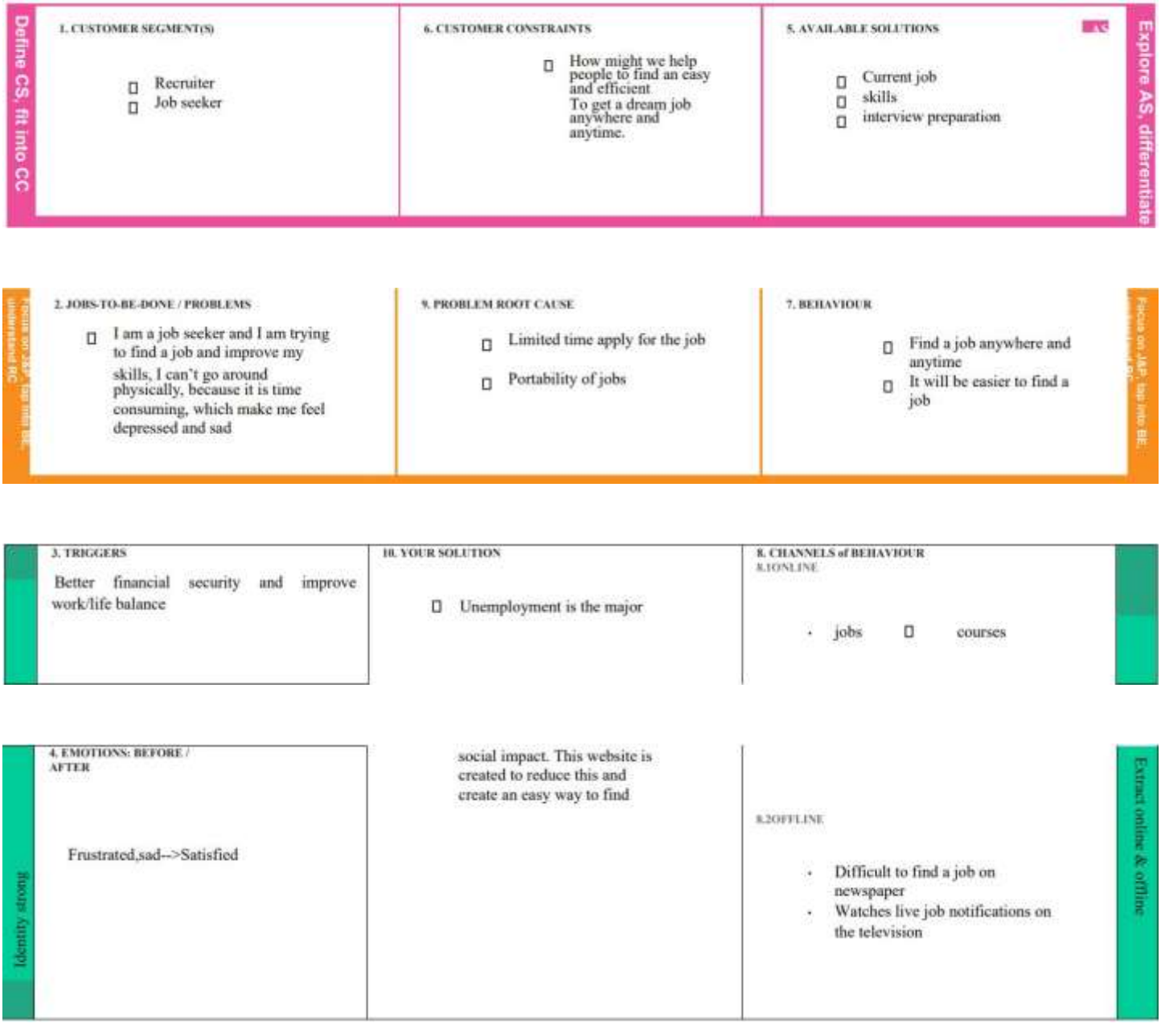
20 minutes



3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Job seeking is very difficult. People spend a lot of time and money searching job. Learning a skill is difficult and finding an institute or a teacher is more difficult. It also costs a fortune to pay for a skill.
2.	Idea / Solution description	Building a website for job seeking and learning skill for free. People can save a lot money and time. This website also gives recommendation according to each person's profile.
3.	Novelty / Uniqueness	It can be done using a lot of other technologies. We are doing this using cloud and flask frame work.
4.	Social Impact / Customer Satisfaction	Unemployment is the major social impact. This website is created to reduce this and creates an easy way to find.
5.	Business Model (Revenue Model)	In the money point of view this project costs less. This includes cloud spaces cost and developers cost. When compared to many other projects it is less.
6.	Scalability of the Solution	Citizens, Companies, Job seekers.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Register using E-mail Register using Mobile phone
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Browse	Browse for job Browse for learning
FR-4	User post	Post job for Job seekers
FR-5	User learn	User can gain skill using this platforms without any payment
FR-6	Recommendation	Recommendation as per their search

4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The UI for this project is user friendly and easy to navigate.
NFR-2	**Security**	Authentication for each system is provided.
NFR-3	**Reliability**	The system must perform without fail for at-least 95% of the time.
NFR-4	**Performance**	The landing page must be very supportive and very supportive.
NFR-5	**Availability**	The service must be uninterrupted.
NFR-6	**Scalability**	Scalability is comparatively high.

5. PROJECT DESIGN

5.1 Data Flow Diagram

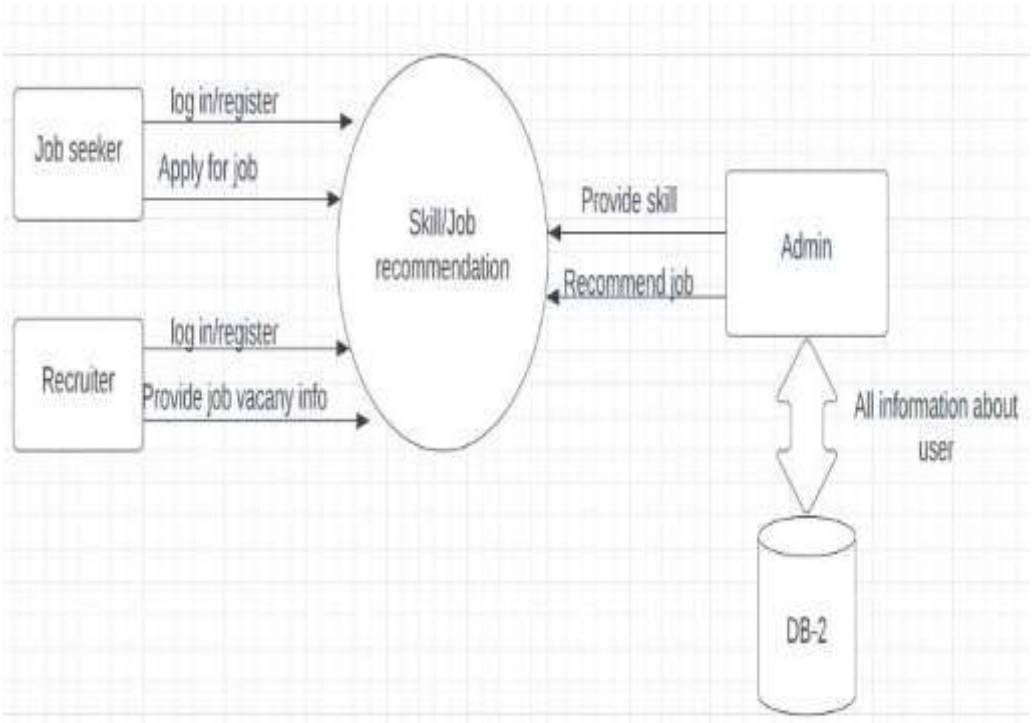


Fig: Data Flow Diagram

5.2 Solution & Technical Architecture

a) Solution Architecture Diagram

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

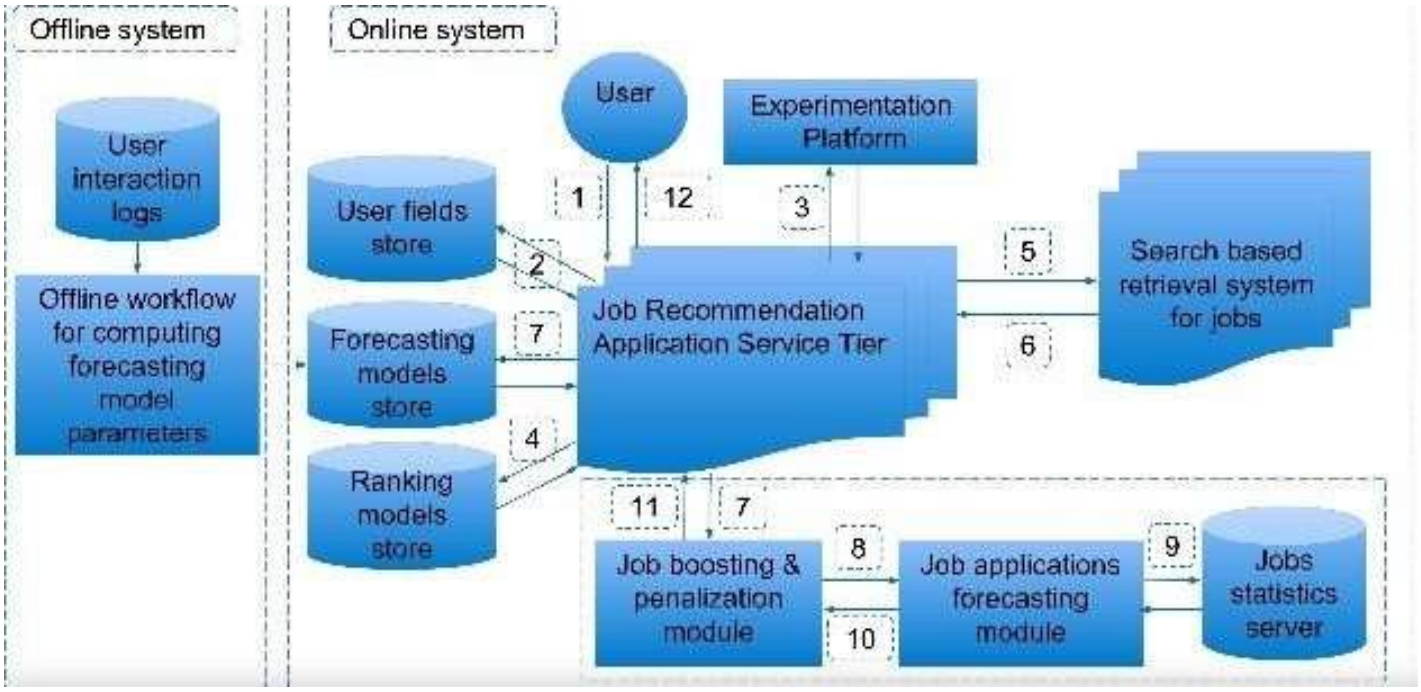
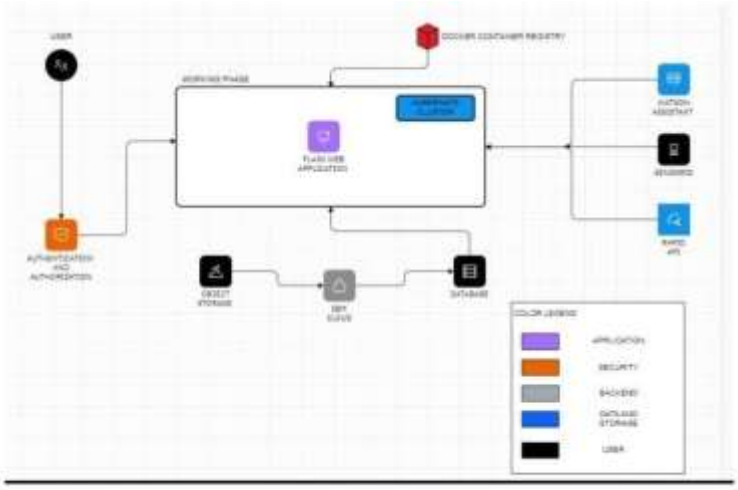


Fig: Architecture and data flow of the job recommender Application

b) Technical Architecture



Guidelines:

1. Registration using form, Gmail
2. Confirmation using OTP, gmail
3. flask app -using python library
4. first - Homepage with login and register
5. In Homepage showing post job and apply job
6. Login page- login and confirmation message
7. register page- register an confirm using OTP or email
8. after Login-dashboard showing more jobs and can search specific job
9. Login and register database are stored in IBM DB2
10. OTP Messages are sent through Send grid
11. Rapid api is connected to display jobs and to search jobs
12. Files can be stored in IBM Storage
13. Services are received from IBM Cloud account

Fig: Technical Architecture

Table 1: Application Characteristics

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework flask.
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used IBM cloud
4.	Availability	Justify the availability of application (e.g., use of load balancers, distributed servers etc.)	Technology used IBM DB2, Kubernetes
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used sendgrid , IBM container Registry,

Table 2: Components and Technologies

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Web UI, Mobile App, Chatbot etc.	HTML, CSS, Java script, flask etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2 (ibm cloud)
7.	File Storage	File storage requirements	IBM object Storage
8.	External API-1	Purpose of External API used in the application	Rapid API API, etc.
9.	Infrastructure (Server / Cloud)	Purpose of External API used in the application	Rapid API, etc.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the website by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the web	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the website through Google	I can register & access the dashboard with Google Login	Low	Sprint-2
		USN-4	As a user, I can register for the website through G-mail.	I can register through Google mail	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the website by entering email & password	I can log in and access my account	High	Sprint-3
	Dashboard	USN-6	As a user, I enter the dashboard and learn new skills.	I can access my dashboard and learn new skill.	High	Sprint-3
Administrator	Recommendation	USN-7	As an administrator, I recommend new job	I can give job recommendation	High	Sprint-4
	Maintenance	USN-8	As an administrator, I can access and store all the details about users in DB-2.	I can access the database	High	Sprint-4

PROJECT PLANNING AND SCHEDULING

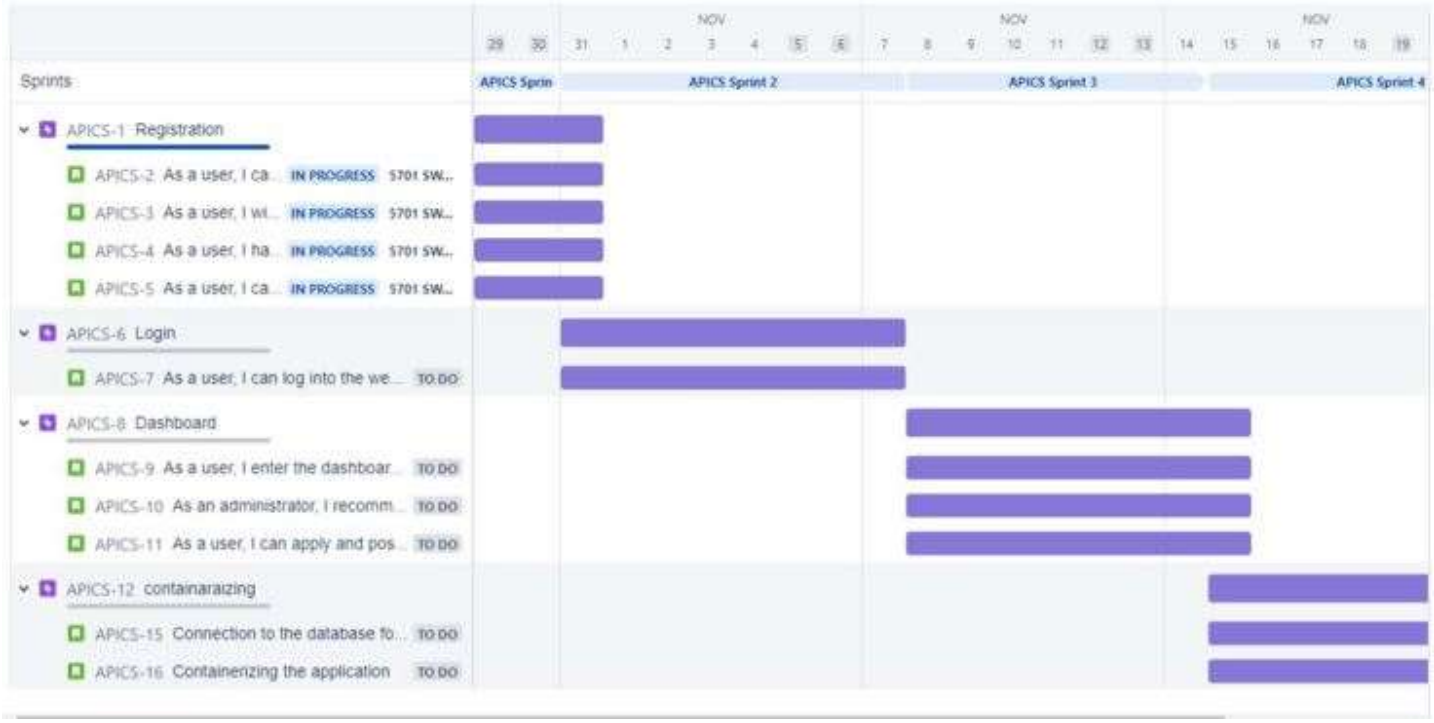
5.4 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the website by entering my email, password, and confirming my password.	High	Dharunraj S
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the web	High	Abishek R
Sprint-2	Webpage Registration	USN-3	As a user, I can register for the website through Google	High	Aakash Veeraiah G
Sprint-2	Mail Registration	USN-4	As a user, I can register for the website through G-mail.	Low	Arunprakash K S
Sprint-3	Login	USN-5	As a user, I can log into the website by entering email & password	Medium	Aakash Veeraiah G
Sprint-3	Dashboard	USN-6	As a user, I enter the dashboard and learn new skills.	Medium	Arunaprakash K S Abishek R
Sprint-4	Recommendation	USN-7	As an administrator, I recommend new job	High	Dharunraj S Aakash Veeraiah G
Sprint-4	Maintenance	USN-8	As an administrator, I can access and store all the details about users in DB-2.	**High**	Dharunaj S Aakash Veeraiah G Abishek R Arunaprakash K S

5.5 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	**11**	**5 days**	1 Nov 2022	6 Nov 2022	**10**
Sprint-2	**4**	**6 days**	6 Oct 2022	11 Nov 2022	**6**
Sprint-3	**8**	**6 days**	11 Nov 2022	17Nov 2022	**7**
Sprint-4	**5**	**6 days**	17 Nov 2022	23 Nov 2022	**5**

5.6 Reports from JIRA



6. CODING AND SOLUTIONING

6.1 Feature 1 – Courses

In the “Skill and Job Recommender” application online courses will be available for the users. These courses are provided by the application itself and are available for free. At least one course will be available for each skill.

The users can access the courses in the skills tab. The courses are in text format. That is, there are no video courses right now. However, links to useful videos can be provided. The user can use these courses to improve their skills.

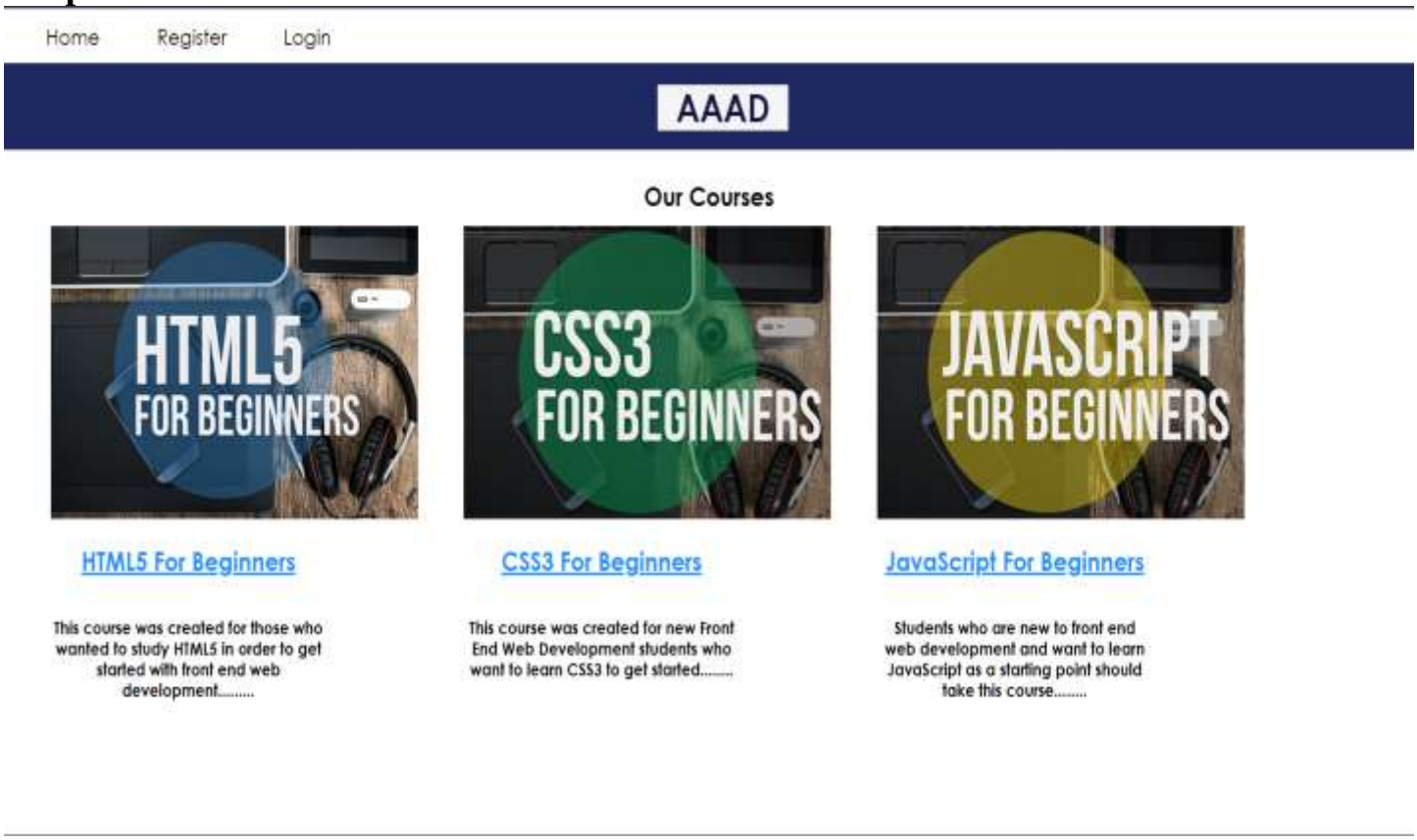
CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AAAD</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('home')}}">Home</a></li>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">AAAD</span></h1></center>
```

```
</section>
   <br><br>
   <section id="courses">

            <center><h1>Our Courses</h1></center>

            <div class="course">
        <img src="{{url_for('static',filename='img/course-01.jpg')}}"alt="skill 1">



                    <center><h2><a href="https://www.w3schools.com/html/">HTML5 For
Beginners</a></h2></center>

                    <center><h6>This course was created for those who wanted to study
HTML5 in order to get started with front end web development.....</h6></center>



            </div>

            <div class="course2">
        <img src="{{url_for('static',filename='img/course-02.jpg')}}"alt="skill 2">


                    <center><h2><a href="https://www.w3schools.com/css/">CSS3 For
Beginners</a></h2></center>

                    <center><h6>This course was created for new Front End Web Development
students who want to learn CSS3 to get started.....</h6></center>



            </div>

            <div class="course3">

                    <img src="{{url_for('static',filename='img/course-03.jpg')}}"alt="skill 3">

                    <center><h2><a href="https://www.w3schools.com/js/">JavaScript For
Beginners</a></h2></center>

                    <center><h6>Students who are new to front end web development and want
```

```
to learn JavaScript as a starting point should take this course.....</h6></center>


                </div>

        </section>

</body>
</html>
```

Output:



6.2 Feature 2 – Expense Overall Report

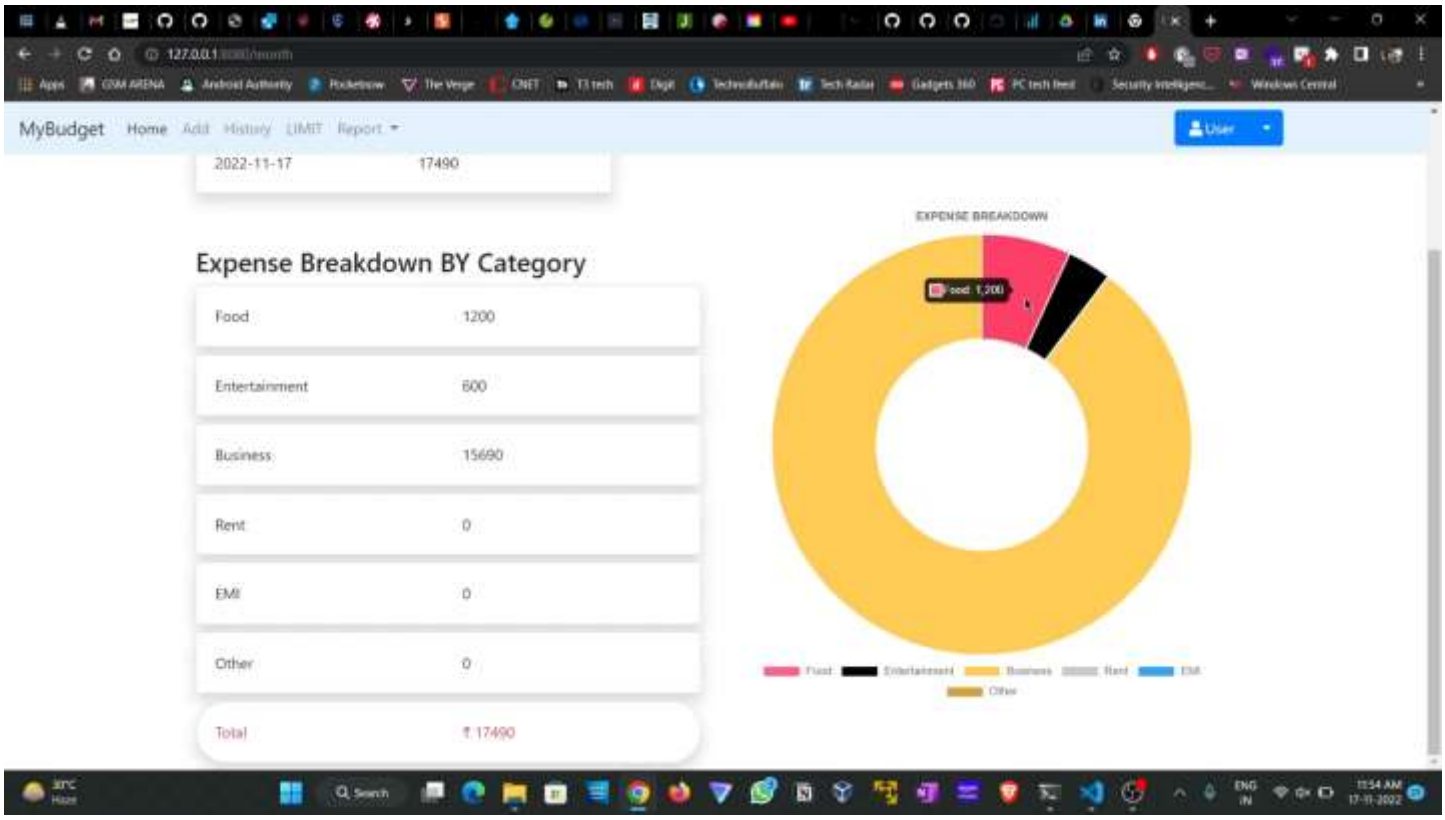
With the aid of user-entered expense data, the overall report will be generated in this application. The application will use the user's input to calculate the daily, monthly, and yearly expenses and display them in a graph format.

Code:

```
@app.route("/year")
def year():


      param1 = "SELECT MONTH(date) as mn, SUM(amount) as tot FROM expenses WHERE userid = " +
str(session['id']) + " AND YEAR(date) = YEAR(current timestamp) GROUP BY MONTH(date) ORDER BY
MONTH(date)"
      res1 = ibm_db.exec_immediate(ibm_db_conn, param1)
      dictionary1 = ibm_db.fetch_assoc(res1)
      texpense = []

      while dictionary1 != False:
          temp = []
          temp.append(dictionary1["MN"])
          temp.append(dictionary1["TOT"])
          texpense.append(temp)
          print(temp)
          dictionary1 = ibm_db.fetch_assoc(res1)



      param = "SELECT * FROM expenses WHERE userid = " + str(session['id']) + " AND
YEAR(date) = YEAR(current timestamp) ORDER BY date DESC"
      res = ibm_db.exec_immediate(ibm_db_conn, param)
      dictionary = ibm_db.fetch_assoc(res)
      expense = []
      while dictionary != False:
          temp = []
          temp.append(dictionary["ID"])
          temp.append(dictionary["USERID"])
          temp.append(dictionary["DATE"])
          temp.append(dictionary["EXPENSENAME"])
          temp.append(dictionary["AMOUNT"])
          temp.append(dictionary["PAYMODE"])
          temp.append(dictionary["CATEGORY"])
          expense.append(temp)
          print(temp)
          dictionary = ibm_db.fetch_assoc(res)


      total=0
```

```
    t_food=0
    t_entertainment=0
    t_business=0
    t_rent=0
    t_EMI=0
    t_other=0


    for x in expense:
        total += x[4]
        if x[6] == "food":
            t_food += x[4]

        elif x[6] == "entertainment":
            t_entertainment  += x[4]

        elif x[6] == "business":
            t_business  += x[4]
        elif x[6] == "rent":
            t_rent  += x[4]

        elif x[6] == "EMI":
            t_EMI  += x[4]

        elif x[6] == "other":
            t_other  += x[4]

    print(total)

    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)



    return render_template("today.html", texpense = texpense, expense = expense,  total =
total ,
                           t_food = t_food,t_entertainment =  t_entertainment,
                           t_business = t_business,  t_rent =  t_rent,
                           t_EMI =  t_EMI,  t_other =  t_other )
```

Output:



6.3 Database Schema

Username: Varchar (32)

Email: Varchar (32)

Phone Number: Varchar (32)

Password: Varchar (32)

Pin: Varchar (32)

6.4 IBM Db2

- A hybrid ANSI-compliant data virtualization tool for accessing, querying and summarizing data across the enterprise which:
- Provides a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities
- Requires only a single database connection or query to connect disparate sources such as HDFS, RDMS, NoSQL databases, object stores and Web HDFS
- Provides low latency support for ad-hoc and complex queries, highperformance, and federation capabilities
- Understands dialects from other vendors and various products fromOracle, IBM® Db2® and IBM Netezza®
- Enables advanced row and column security

KUBERNETES

- Kubernetes — also known as “k8s” or “kube” — is a container orchestration platform for scheduling and automating the deployment, management, and scaling of containerized applications.
- Kubernetes was first developed by engineers at Google before being open sourced in 2014. It is a descendant of Borg, a container orchestration platform used internally at Google. Kubernetes is Greek
- Today, Kubernetes and the broader container ecosystem are maturing into a general-purpose computing platform and ecosystem that rivals — if not surpasses — virtual machines (VMs) as the basic building blocks of modern cloud infrastructure and applications.
- This ecosystem enables organizations to deliver a high- productivity Platform-as-a-Service (PaaS) that addresses multiple infrastructure-related and operations-related tasks and issues
- surrounding cloud-native development so that development teams can focus solely on coding and innovation.

7. TESTING

7.1 Test Cases

- Verify that after registration users are navigated to login page
- Verify the Ul elements in Login/Signup popup
- Verify user is able to log into application with Valid credentials
- Verify that categories of news are shown in homepage
- Verify that news is displayed in homepage
- Verify that when clicked on news it is redirected to correct page

7.2 User Acceptance Testing

Purpose of Document

The goal of this document is to explain briefly the test coverage and unresolved issues of the Skills/Job Recommender Application project at the time of delivery to User Acceptance Testing (UAT).

Defect Analysis

The report shows the number of resolved or closed errors at each severity level, and how they are solved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	2	2	1	8
Duplicate	1	0	3	0	4
External	2	0	0	1	3
Fixed	5	2	4	7	18
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	11	9	13	11	44

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	4	0	0	4
Security	3	0	0	3
Outsource Shipping	7	0	0	7
Exception Reporting	6	0	0	6

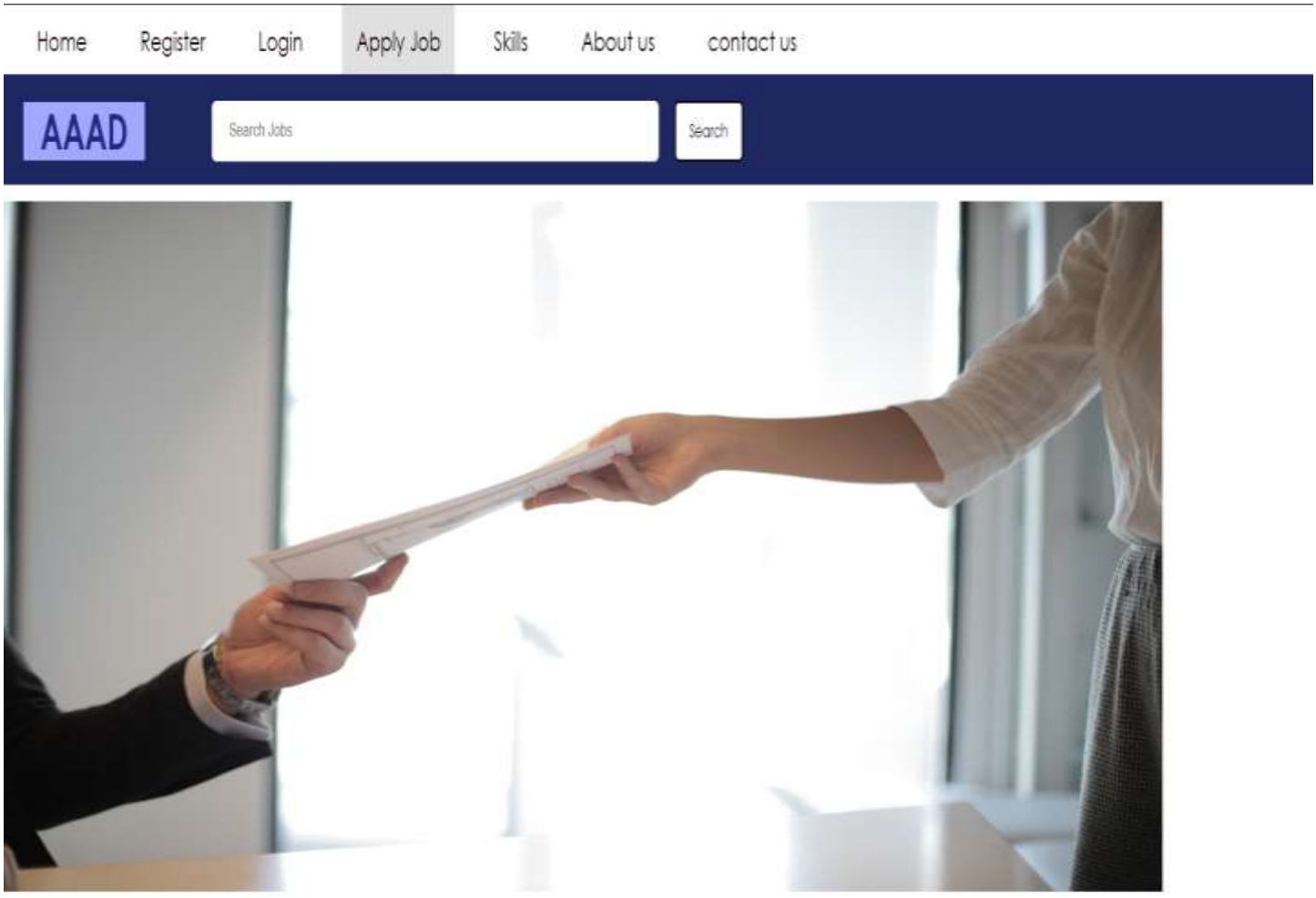
8. RESULT

8.1 Performance Metrics

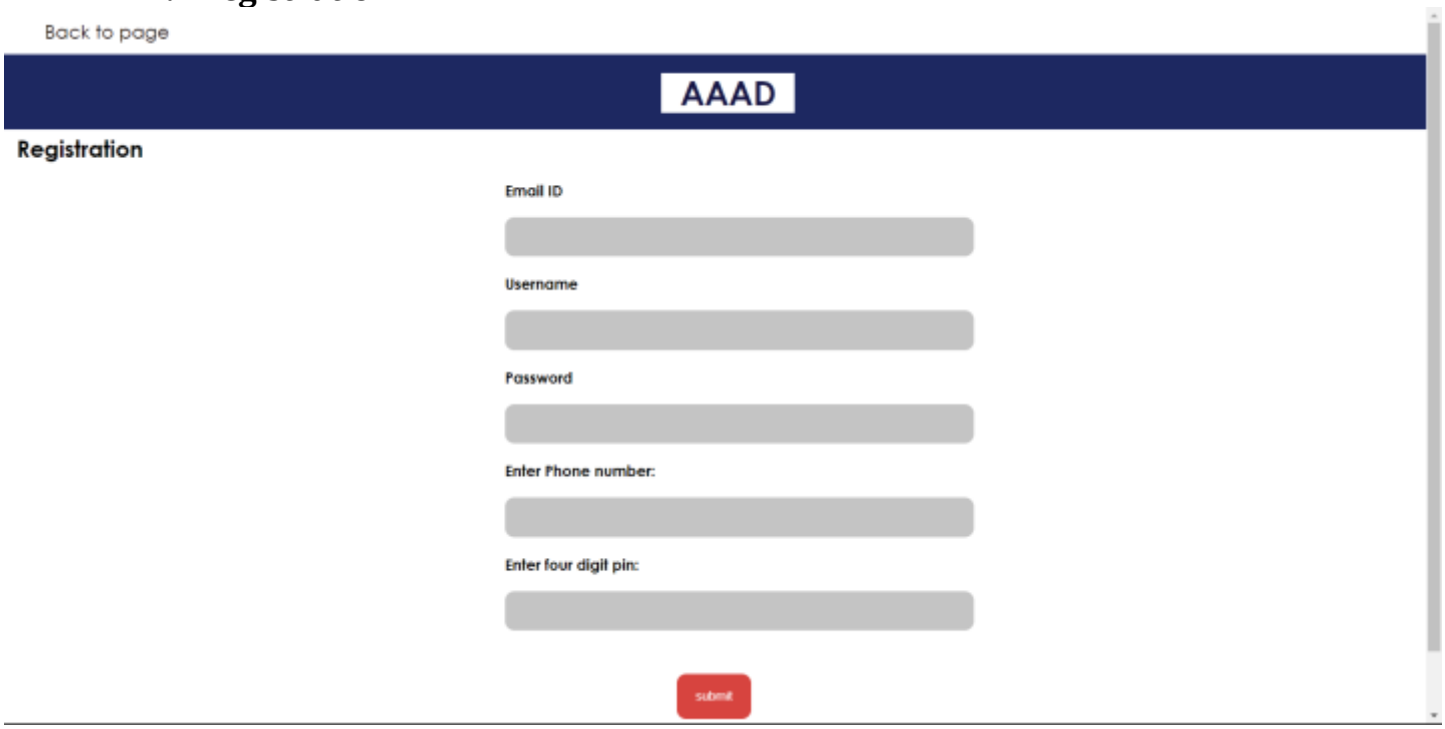
The application performance index, or Apdex score, has become an industry standard for tracking an application's relative performance. It works by setting a time limit for how long a certain online request or transaction should take. These transactions are then classified as satisfied (quick), tolerating (slow), too slow, or failed. A basic math method is then used to get a score ranging from O to 1

Screenshots:

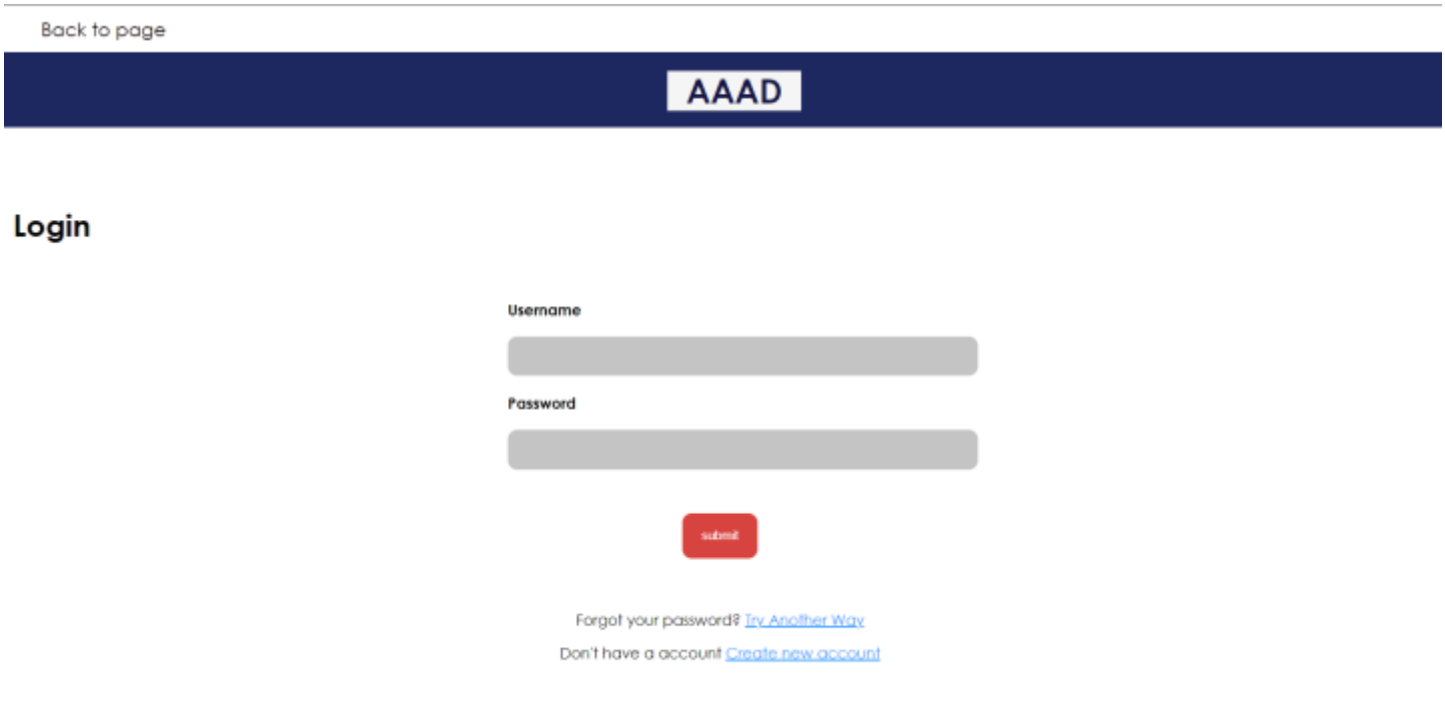
1.Homepage:



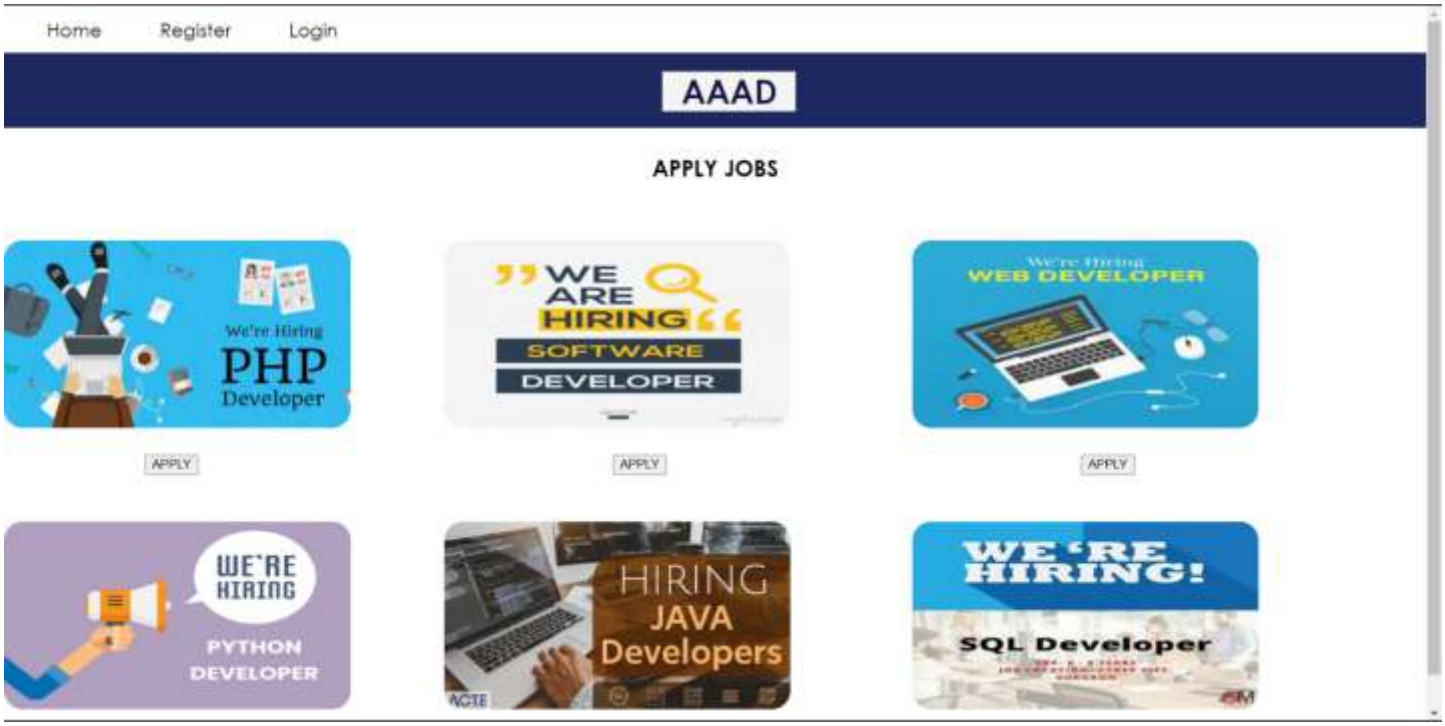
2. Registration



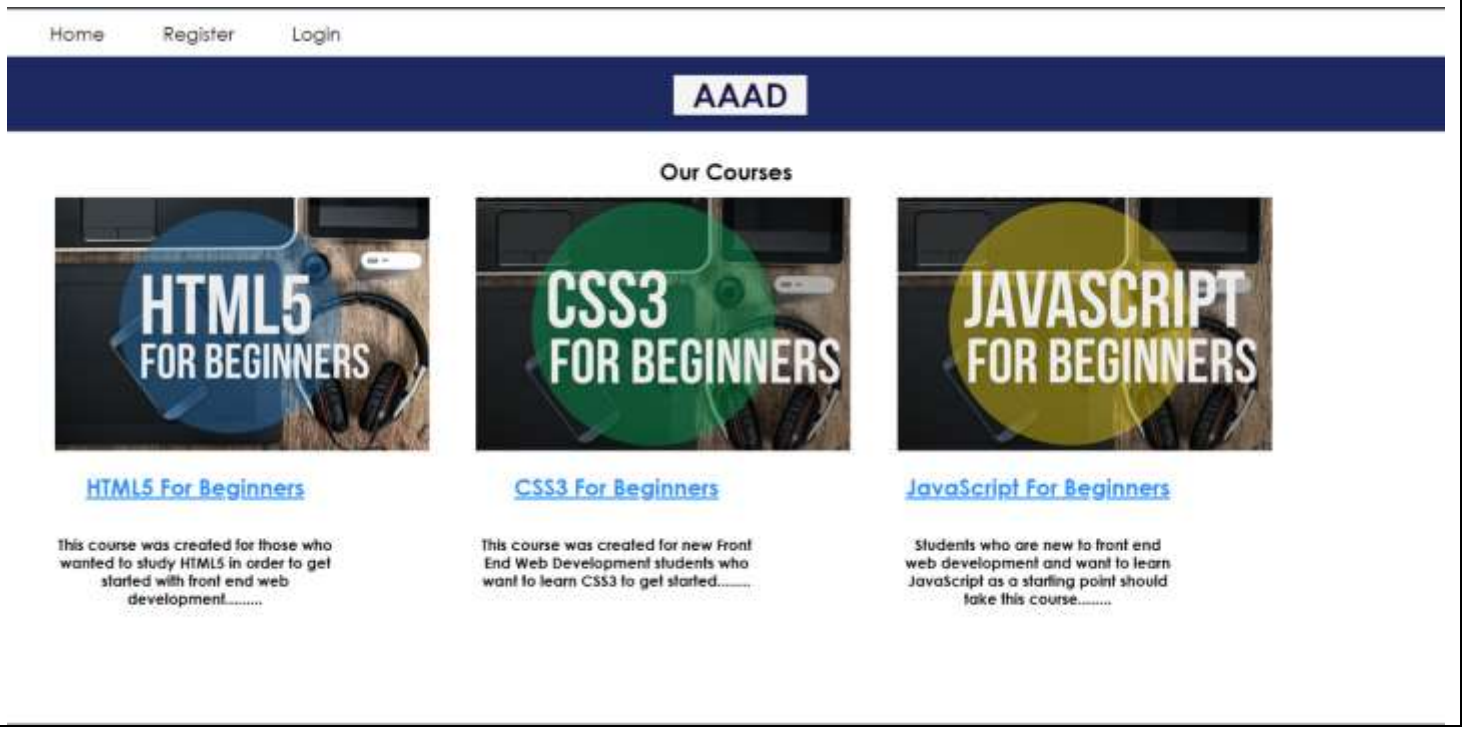
3. Login Page



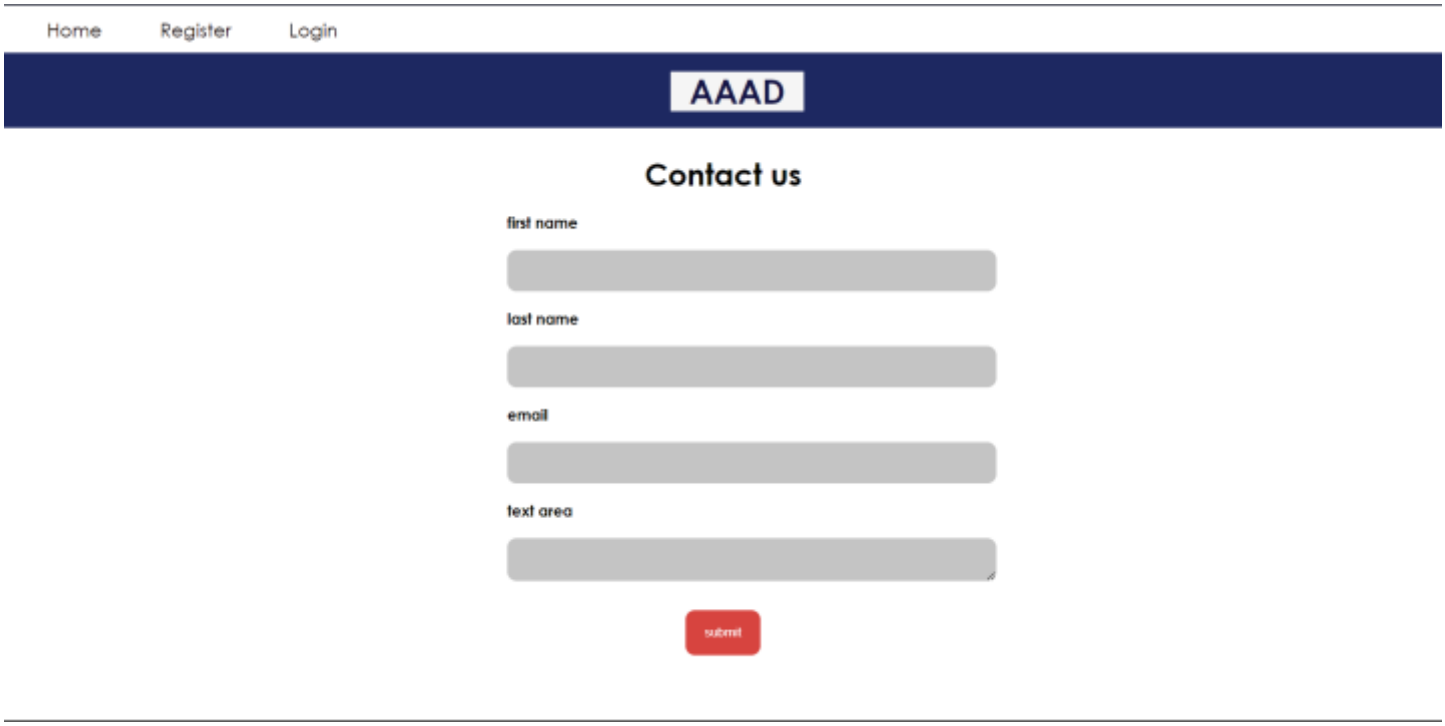
4. Apply Job



5. Skill Recommender



6. Contact



7. About

9. ADVANTAGES AND DISADVANTAGES

a) ADVANTAGES

1.**Employment Possibilities:**

The primary benefit of having a profile in our application is that it opens the door to job prospects all around the world. Prior to the emergence of internet job applications, students obtained employment through personal contacts. However, your work opportunities have really expanded recently. Students who have completed their study overseas can enter their field of specialization and find suitable employment. Aside from that, if there is a certain company in which you are interested, you can apply for it.
to 1.

2. **Simple job applications**

Online job applications are now more important than the conventional hiring process, which has taken a backseat. The time when you had to carry copies of your résumé with you everywhere are long gone. The simplicity of adding the necessary data to your profile means that you can update your skills frequently in addition to having recruiters look through it. The fact that the recruiter is already aware of your skills and wants to further explore them lessens the initial stress of a job application.

3**. Initiate Connections**:

Apart from receiving a job offer, the connections you establish on your profile help you in the long run. For instance, you may start by connecting with your school and college friends and eventually shift to your colleagues. An alumnus from your university is good connections to have. Having an illustrious list of connections speaks to your strong.

b) DISADVANTAGES

1. Risk of Identity Theft

You must include a huge amount of personal information in your profile for the eyes of potential employers. Therefore, if LinkedIn servers have a problem, you run the danger of losing sensitive information to the public, which could lead to identity theft

2. Incomplete Profile Challenge

LinkedIn like other social network websites required you to put up an attractive profile. That is a profile that is appealing to employers and prospective recruiters. People however find it hard to fill out profile details.

3. Premium Package cab be Expensive

It's a good thing they don't come cheap. LinkedIn, on the other hand, allows you to join the platform for free. However, the LinkedIn premium packages are not free.
For example, the "medium-sized career" fee is approximately $29.99 each month. This provides numerous additional benefits. Provides, but it might still be too expensive for a small or medium-sized organization.

4. Tons of Spam Messages

There is a saying that states there will always be a Judas among the 12 disciples. Consider the number of Judas that will be accessible on a website with more than 1200 million users. Spam communications from recruiters, businesses, and even job seekers are abundant on LinkedIn. All only meant to attract attention, deceive, and to demand money, etc.

10. CONCLUSION

We proposed a job suggestion application. This programme simplifies the job suggestion process by allowing the use of a number of text processing and recommendation methods based on the preferences of the job recommender system creator. Furthermore, we contribute by making public a new dataset featuring job seeker profiles and job openings. Future directions of our study will center on doing a more thorough evaluation that takes into account a larger number of methodologies and data, as well as a full examination of the impact of each professional skill of a job seeker on the obtained job recommendation.

11. FUTURE SCOP

12. APPENDIX

12.1 Source Code

App.py

```
from flask import Flask,
render_template,request,redirect,url_for,session,flash
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import requests

app=Flask(__name__)
app.secret_key='a'
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-
bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304
;SECURITY=SSL;SSLServerCertificat=DigiCertGlobalRootCA.crt;UID=wkz1989
6;PWD=ANpZE9kHEdlNhNgE",'','')
except:
    print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
def dash():
    return render_template('home.html',msg=" ")

@app.route("/register",methods=['GET','POST'])
def register():
    error = None
    if request.method=='POST':
            username=request.form['username']
            email=request.form['email']
            phone_number=request.form['phonenumber']
            password=request.form['password']
            pin=request.form['pin']
            sql="SELECT * FROM user WHERE phone_number=?"
            prep_stmt=ibm_db.prepare(conn,sql)
            ibm_db.bind_param(prep_stmt,1,phone_number)
```

```
            ibm_db.execute(prep_stmt)
            account=ibm_db.fetch_assoc(prep_stmt)
            print(account)
              #message =
Mail(from_email='btechmano@gmail.com',to_emails=session['email'],subje
ct="Devnews - Registration",html_content='<b>Devnews welcomes
you</b><br/><p>Your account has been registered successfully</p>')
             #try:
                 #sg=SendGridAPIClient()
                 # Secret key can't be submitted otherwise my
                 # sendgrid accound reporting that i am exposing
                 # my secret key as public and my account will
terminated soon
                 #response=sg.send(message)
                 #print(response.status_code)
                 #print(response.body)
                 #print(response.headers)
             #except Exception as e:
                 #print(e)
            if account:
                error="Account already exists! Log in to continue !"
            else:
                insert_sql="INSERT INTO user values(?,?,?,?,?)"
                prep_stmt=ibm_db.prepare(conn,insert_sql)
                ibm_db.bind_param(prep_stmt,1,email)
                ibm_db.bind_param(prep_stmt,2,username)
                ibm_db.bind_param(prep_stmt,3,phone_number)
                ibm_db.bind_param(prep_stmt,4,password)
                ibm_db.bind_param(prep_stmt,5,pin)
                ibm_db.execute(prep_stmt)
                flash(" Registration successfull. Log in to continue
!")
    else:
        pass
    return render_template('register.html',error=error)

@app.route('/login',methods=['GET','POST'])
def login():
    error = None
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']
```

```
        sql="SELECT * FROM user WHERE username=? AND password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / password"
            return render_template('login.html',error=error)
    return render_template('login.html',error=error)

@app.route('/forget',methods=['GET','POST'])
def forget():
    error = None
    if request.method=='POST':
        username=request.form['username']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE username=? AND pin=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,pin)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / pin"
            return render_template('login.html',error=error)
    return render_template('forget.html',error=error)
@app.route('/welcome')
```

```
def welcome_page():
    return render_template("welcome.html",msg=" ")
@app.route('/home')
def home():
    return render_template("home.html",msg=" ")
@app.route('/skills')
def skills():
    return render_template("skills.html",msg=" ")
@app.route('/about')
def about():
    return render_template("about.html",msg=" ")
@app.route('/contact')
def contact():
    return render_template("contact.html",msg=" ")
if __name__=='__main__':
    app.run(debug=True)
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>AAAD</title>
   <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
   <script>
      window.watsonAssistantChatOptions = {
         integrationID: "3VT2zCBlLAkCSx6T3ZtJyxc-VWilfCq9SUIZLawuBjr7", // The ID of
this integration.
         region: "us-south", // The region your integration is hosted in.
         serviceInstanceID: "efd0a9c4-8cc9-44c8-81a3-316adb81ab19", // 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>
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
        <li><a href="{{url_for('welcome_page')}}">Apply Job</a></li>
        <li><a href="{{url_for('skills')}}">Skills</a></li>
        <li><a href="{{url_for('about')}}">About us</a></li>
        <li><a href="{{url_for('contact')}}">contact us</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <h1 class="two" ><span style="color:rgb(37, 37, 133);background-color: rgb(161, 169,
255);margin-left: 30px;padding-left: 20px;padding-right: 20px;">AAAD</span></h1>
    <div id="ss1">
      <input class="input_search" type="text" placeholder="Search Jobs" >
      <button class="button_search">Search</button>
    </div>
    <img src="{{url_for('static',filename='img/bg.png')}}"alt="job" height="500"
width="1360">
    </section>
</body>
</html>
```

Login.html

```
<html>
  <head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
      </ul>
    </nav>
```

```
    </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">AAAD</span></h1></center>
    </section>
    <br><br>
    <div class="container"> <br /><br />
    <form action="/login" method="POST">
     <h1>Login</h1> <br /><br />
      <label class="form_label"for="email"><b>Username</b></label><br><br>
      <input class="form_input"type="text" name= "username" /><br><br>
      <label class="form_label"for="psw"><b>Password</b></label><br><br>
      <input class="form_input"type="password" name="password"/>
      </br></br></br>
      <center><input type="submit" class="submitbtn"value="submit" /></center>
    </form>
    </div>
    <center><p>Forgot your password? <a href="{{url_for('forget')}}">Try Another
Way</a></p></center>
    <center><p>Don't have a account <a href="{{url_for('register')}}">Create new
account</a></p></center>
  </body>
</html>
```

Register.html

```
<html>
  <head>
    <title>Registration page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
      </ul>
    </nav>
  </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-
color:white;margin-left: 30px;padding-left: 20px;padding-right:
```

```
20px;">AAAD</span></h1></center>
      </section>
      <div class="container">
      <form action="/register" method="POST">
        <h2>Registration</h2>
        <label class="form_label"for="email"><b>Email ID</b></label><br><br>
        <input class="form_input" type="email" name="email"/><br><br>
        <label class="form_label"for="user"><b>Username</b></label><br><br>
        <input class="form_input"type="text" name= "username" /><br><br>
        <label class="form_label"for="psw"><b>Password</b></label><br><br>
        <input class="form_input"type="password" name="password"/><br><br>
        <label class="form_label"for="pho"><b> Enter Phone number:</b></label><br><br>
        <input class="form_input"type="text" name="phonenumber"/><br><br>
        <label class="form_label"for="pho"><b> Enter four digit pin:</b></label><br><br>
        <input class="form_input"type="password" name="pin"/>
        </br></br></br>
       <center> {% if error %}
      <p><strong style="color:red">Error</strong>: {{error}}</p>
   {% endif %}
   {% with messages = get_flashed_messages() %}
       {% if messages %}
           {% for message in messages %}
               <p style="color:green">{{ message }}</p>
           {% endfor %}
       {% endif %}
    {% endwith %} </center>
        <center><input type="submit" class="submitbtn"value="submit" /></center>
        <center><p>Already have a account <a href="{{url_for('login')}}">Sign
in</a></p></center>
      </form>
      </div>
   </body>
</html>
```

Skills.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>AAAD</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('home')}}">Home</a></li>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">AAAD</span></h1></center>
    </section>
    <br><br>
    <section id="courses">

                <center><h1>Our Courses</h1></center>

                <div class="course">
            <img src="{{url_for('static',filename='img/course-01.jpg')}}"alt="skill 1">



                        <center><h2><a href="https://www.w3schools.com/html/">HTML5 For
Beginners</a></h2></center>

                        <center><h6>This course was created for those who wanted to study
HTML5 in order to get started with front end web development.....</h6></center>



                </div>

                <div class="course2">
            <img src="{{url_for('static',filename='img/course-02.jpg')}}"alt="skill 2">
```

```
                <center><h2><a href="https://www.w3schools.com/css/">CSS3 For Beginners</a></h2></center>

                <center><h6>This course was created for new Front End Web Development students who want to learn CSS3 to get started.....</h6></center>



            </div>

            <div class="course3">

                <img src="{{url_for('static',filename='img/course-03.jpg')}}"alt="skill 3">

                <center><h2><a href="https://www.w3schools.com/js/">JavaScript For Beginners</a></h2></center>

                <center><h6>Students who are new to front end web development and want to learn JavaScript as a starting point should take this course.....</h6></center>



            </div>

        </section>

</body>
</html>
```

Welcome.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>AAAD</title>
   <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
   <header>
```

```
        <nav class="navbar">
          <ul>
             <li><a href="{{url_for('home')}}">Home</a></li>
            <li><a href="{{url_for('register')}}">Register</a></li>
            <li><a href="{{url_for('login')}}">Login</a></li>
          </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">AAAD</span></h1></center>
    </section>
    <br><br>
    <section id="jobs">

                <center><h1><b>APPLY JOBS</b></h1></center><br>
        <img style="position:absolute;top: 250px;border-radius:
20px;"src="{{url_for('static',filename='img/php.jpg')}}" alt=""width="370" height="200">
        <img style="position:absolute;top: 550px;border-radius:
20px;"src="{{url_for('static',filename='img/python.png')}}" alt=""width="370" height="200">
        <img style="position:absolute;top: 250px;left: 470px;border-radius:
20px;"src="{{url_for('static',filename='img/software.jpg')}}" alt=""width="370"
height="200">
        <img style="position:absolute;top: 550px;left: 470px;border-radius:
20px;"src="{{url_for('static',filename='img/java.jpg')}}" alt=""width="370" height="200">
                <img style="position:absolute;top: 250px;left: 970px;border-radius:
20px;"src="{{url_for('static',filename='img/web.jpg')}}" alt=""width="370" height="200">
        <img style="position:absolute;top: 550px;left: 970px;border-radius:
20px;"src="{{url_for('static',filename='img/sql.jpg')}}" alt=""width="370" height="200">

                <h2 style="position:absolute;top: 450px;left:150px"><a
href="https://in.indeed.com/PHP-Developer-
jobs?vjk=10cca9575b193c7d"><button>APPLY</button></a></h2>
        <h2 style="position:absolute;top: 450px;left:650px"><a
href="https://in.indeed.com/Software-Developer-
jobs?vjk=b7da08f07cac87d5"><button>APPLY</button></a></h2>
        <h2 style="position:absolute;top: 450px;left:1150px"><a
href="https://in.indeed.com/Web-Developer-
jobs?vjk=b81e49165da51eeb"><button>APPLY</button></a></h2>
        <h2 style="position:absolute;top: 750px;left:1150px"><a
href="https://in.indeed.com/SQL-Developer-
jobs?vjk=ac86b15908022123"><button>APPLY</button></a></h2>
```

```
        <h2 style="position:absolute;top: 750px;left:650px"><a href="https://in.indeed.com/Java-Developer-jobs?vjk=da306a665e00eb30"><button>APPLY</button></a></h2>
        <h2 style="position:absolute;top: 750px;left:150px"><a href="https://in.indeed.com/Python-Developer-jobs?vjk=fa7b9bd250044569"><button>APPLY</button></a></h2>
    </section>
</body>
</html>
```

Forget.html

```
<html>
    <head>
        <title>Login page</title>
        <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
    </head>
    <body>
        <header>
        <nav class="navbar">
            <ul>
              <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
            </ul>
        </nav>
        </header>
        <section id="sss1">
            <center><h1 class="two" ><span style="color:rgb(109, 30, 192);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">AAAD</span></h1></center>
        </section>
        <br><br>
    {% if error %}
        <p><strong style="color:red">Error</strong>: {{error}}</p>
    {% endif %}
    {% with messages = get_flashed_messages() %}
        {% if messages %}
            {% for message in messages %}
                <p style="color:green">{{ message }}</p>
            {% endfor %}
        {% endif %}
      {% endwith %}
        <div class="container"> <br /><br />
        <form action="/forget" method="POST">
```

```
        <h1>Try to login with your 4 digit pin</h1> <br /><br />
          <label class="form_label"for="email"><b>Username</b></label><br><br>
          <input class="form_input"type="text" name= "username" /><br><br>
          <label class="form_label"for="psw"><b>Pin</b></label><br><br>
          <input class="form_input"type="password" name="pin"/>
          </br></br></br>
          <center><input type="submit" class="submitbtn"value="submit" /></center>
       </form>
       </div>
       <center><p>Back to <a href="{{url_for('login')}}">login</a></p></center>
   </body>
</html>
```

About.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>AAAD</title>
   <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
   <header>
      <nav class="navbar">
         <ul>
            <li><a href="{{url_for('home')}}">Home</a></li>
           <li><a href="{{url_for('register')}}">Register</a></li>
           <li><a href="{{url_for('login')}}">Login</a></li>
         </ul>
      </nav>
   </header>
   <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">AAAD</span></h1></center>

   </section>
   <br>
   <br>
```

```
    <center><h1><b>ABOUT US
    </b></h1></center>
    <P> <b> <h3>Hello to everybody! This project was completed for IBM Nalaiyathiran. Four people make up our team. Skill/Job Recommender Application is our project. This project suggests both talents and Jobs for Job Seekers. We sincerely appreciate IBM for offering their free resources, and we learned a lot about new technology as a result.</h3></b></P>
<h1>TEAM : </h1>
<H3>TEAM LEADER   : DHARUNRAJ S</H3>
<H3>TEAM MEMBER 1 : AAKASH VEERAIAH G</H3>
<H3>TEAM MEMBER 2 : ARUNAPRAKASH K S</H3>
<H3>TEAM MEMBER 3 : ABISHEK R</H3>
</body>
</html>
```

Contact.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AAAD</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('home')}}">Home</a></li>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70) ;background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">AAAD</span></h1></center>
    </section>
    <br><br>
    <center><h1 class="h1lt">Contact us</h1></center>
```

```
   <form>

      <label class="form_label"><b>first name</b></label><br><br>
      <input class="form_input" type="text" name="firstname" value="">
 <br><br>

 <label class="form_label"><b>last name</b></label><br><br>
 <input class="form_input" type="text" name="lastname" value="">

 <br><br>

 <label class="form_label"><b>email</b></label><br><br>
 <input class="form_input" type="text" name="email" value="">
 <br>
 <br>

 <label class="form_label"><b>text area</b></label><br><br>
 <textarea class="form_input" type="text" name="text area" value=""></textarea>
 <br><br>
 <center><input type="submit" class="submitbtn"value="submit" /></center>
</form>

</body>
</html>
```

Style.css

```
body
{
   font-family: Century Gothic;
   margin: 0%;
}
ul
{
   list-style-type: none;
   margin: 0%;
   padding: 0;
   overflow: hidden;
}
li
{
```

```
    margin-left: 30px;
    display: inline;
    float: left;
}
li a
{
  display: block;
  color: hsl(0, 71%, 3%);
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
header
{
    width:100%;
    height: 100px;
    background-color:white(240, 240, 234);
    font-size: 20px;
}
li:hover
{
    background-color:#E0E0E0;
}

.two
{
    font-size: 35px;
    margin-top: 20px;
}
#sss1
{
    top: 50px;
    width: 100%;
    height: 80px;
    background-color: rgb(29,40,97);
    position: absolute;
}
.sec_nav
{
    margin-left: 900px;
}
.input_search
```

```
{
    margin-top: -73px;
    margin-left: 250px;
    width: 500px;
    height: 42px;
    border-radius: 5px;
    border: none;
    position: absolute;
    padding-left: 20px;
}
.button_search
{
    position: absolute;
    margin-top: -73px;
    height: 44px;
    margin-left: 790px;
    width: 80px;
    background-color:white;
    border-radius: 5px;
    font-family: Century Gothic;
}
.button_search:hover
{
    background-color:#E0E0E0;
    color: whitesmoke;
}
footer
{
    height: 30px;
    background-color: rgb(255, 63, 63);
    width: 100%;
}
.head {
    height: 50px;
    width: 100%;
    color: whitesmoke;
    background-color: black;
    font-size: 30px;
    font-weight: bold;
    padding-left: 20px;
    padding-top: 10px;
}
```

```
.head-git{
    height: 40px;
    width: 100%;
    justify-content: center;
    align-items: center;
    background-color: rgb(250, 244, 244);
    display: flex;
    flex-flow: row;
    margin-bottom: 5px;
  }
  .head-git h2 {
    font-size: 18px;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    color:black;
  }
  .head-git h2 a {
    text-decoration: none;
    color:blue;
    font-size: 23px;
  }
  .head-git h2 a:hover {
    color:rgb(211, 47, 41);
    background-color: beige;
  }

  .wrapper {
    height: 200px;
    width: 100%;
    margin-left:150px;
    display: flex;
    padding: 10px;
}
.wrapper .item {
    height: 200px;
    min-width: 20%;
    border-radius: 20px;
    margin: 10px;
    box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.4);
    padding: 0.5px;
    border-bottom: 1px black;
}
.wrapper .item img {
```

```
    height: 100%;
    width: 100%;
}
.c-box {
    height: content;
    max-width: 50%;
    display: flex;
    padding: 10px;
    flex-flow: column;
    padding: 1px;
}
.c-box .content {
    height: content;
    min-width: 30%;
    display: flex;
    flex-flow: row wrap;
    margin-left: 130px;
    text-decoration: none;
}
.c-box .content p {
    color: black;
    font-size: 15px;
    font-weight: bold;
}
.c-box .content h2 {
    color: black;
    font-size: 20px;
    font-weight: bold;
}
.head-f{
    height: 40px;
    width: 100%;
    justify-content: center;
    border-top: 2px solid black;
    display: flex;
    flex-flow: column;
  }
  .head-f h2 {
    font-size: 15px;
    color:black;
  }
```

```
/*register and login css*/

/* Add padding to containers */
.container {
  padding: 16px;
  background-color:white;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit button */
.submitbtn {
  background-color: rgb(211, 47, 41);
  color: white;
  padding: 16px 20px;
  margin: 8px ;
  border: none;
  cursor: pointer;
  opacity: 0.9;
  border-radius: 10px;
}

.submitbtn:hover {
  opacity: 1;
}

/* Add a blue text color to links */
a {
  color: dodgerblue;
}
.form_label
{
  margin-left: 35%;
}
.form_input
{
  margin-left: 35%;
  width: 500px;
```

```
  height: 42px;
  padding-left: 20px;
  border-radius: 10px;
  border: none;
  background-color:rgb(196, 196, 196);
}


input {
  display: inline-block;
}

div.course {
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 200px;
  left:50px;
}

div.course2 {
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 200px;
  left: 500px;
}

div.course3 {
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 200px;
  left: 950px;
}
div.jobs1{
```

```
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 200px;
  left:200px;
}
div.jobs2{
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 200px;
  left:700px;
}
div.jobs3{
  width: 300px;
  height: 300px;
  background: white;
  border: 1px solid transparent;
  position: absolute;
  top: 600px;
  left:200px;
}


h6 {
font-size: 1em;
}
```

12.2 GitHub And Project Demo Link

a) GitHub

b) Project Demo Link