PROJECT REPORT

SKILL/JOB RECOMMENDER

Submitted by

PNT2022TMID20801

Annamalai P	–	412419104009
Karthick Raja V	–	412419104050
Yogesh K	–	412419104152
Kishore R	–	412419104054

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

There has been a sudden boom in the technical industry and an increase in the number of good startups. Keeping track of various appropriate job openings in top industry names has become increasingly troublesome. This leads to deadlines and hence important opportunities being missed.

1.2 PURPOSE

Through this research paper, the aim is to automate this process to eliminate this problem. The intention is to aggregate and recommend appropriate jobs to job seekers, especially in the engineering domain. The entire process of accessing numerous company websites hoping to find a relevant job opening listed on their career portals is simplified.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

There has been a sudden boom in the technical industry and an increase in the number of good startups. Keeping track of various appropriate job openings in top industry names has become increasingly troublesome. This leads to deadlines and hence important opportunities being missed.

2.1 REFERENCES

"Students / Job seekers find their desired job based on their Skillset"

Description:

The Internet-based recruiting platforms become a primary recruitment channel in most companies. The recommender system technology aims to help users in finding items that match their personnel interests. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

“Integrating Intelligent CHATBOT for Job recommendation application”

Description:

A Chatbot is a software application that replaces a live human agent to conduct a conversation via text or text to speech. In this system, we demonstrate a chatbot that uses Artificial Intelligence to produce dynamic responses to online client enquiries. This web-based platform provides a vast intelligent base that can help humans to solve problems. The Chatbot recognizes the user's context, which prompts an intended response. Its objective is to reduce human dependency in every organization and reduce the need for different systems for different processes.

"A Study of LinkedIn as an Employment Tool for Job Seeker & Recruiter"

Description:

LinkedIn has become one of the most known social networking portals in terms of global professional connections, networking, job postings, hiring and much more in relevance to employment opportunities. This research was an attempt to identify the utility of Linked in on selection and recruitment. Also, this study has taken the employers' and the prospective candidates for job and employees' perspective, including factors such as recruitment, selection, job opportunities, internal official communication on Linked-in, professional networking, ease of access, less expensive communication tool etc.

"CLOUD STORAGE AND SHARING SERVICES"

Description:

To create a web application that sends files from one email to another email using the SMTP protocol, which is handled in a server-based application. The main advantage of the project in this paper is that it provides a safe, reliable and excellent tool for sharing files in any format. Also, it has infinite scaling capabilities. With a bit of tweak in the code, it can be scaled to handle heavy file loads. The Cloud-based file sharing approach is proposed to provide the following services for external data confidentiality, secure data sharing within the group, protect data from unauthorized access of officials within the group and provide time and number of file accessto users.Whenever information sharing among a bunch arise the file owner sends the user uploads the file on the application and then shares it using the send API. This creates a safe medium of sharing of files and user in control of the data in the whole process of sharing the files

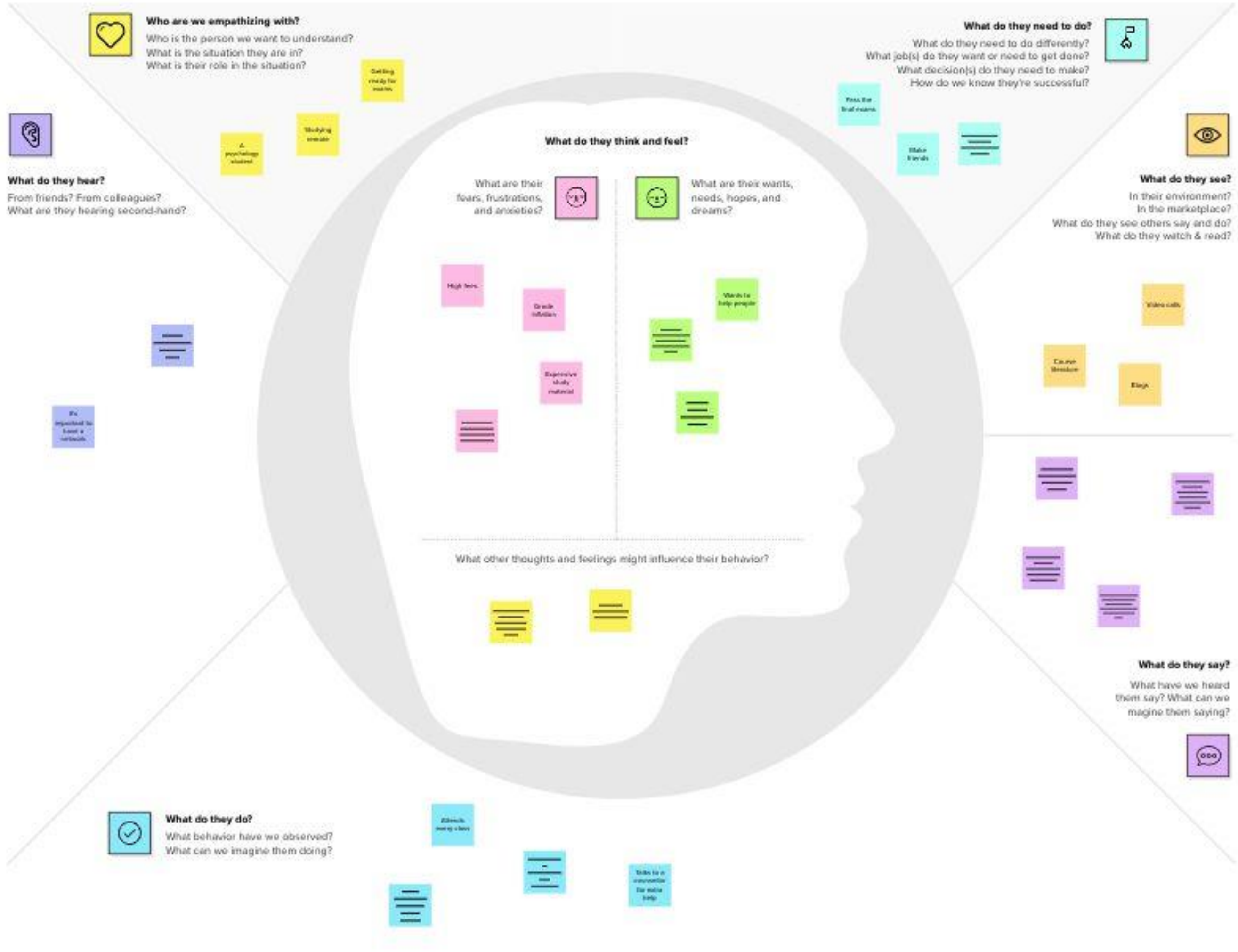
2.2 PROBLEM STATEMENT DEFINITION

The intention is to aggregate and recommend appropriate jobs to job seekers, especially in the engineering domain. The entire process of accessing numerous company websites hoping to find a relevant job opening listed on their career portals is simplified. The proposed recommendation system is tested on an array of test cases with a fully functioning user interface in the form of a web application. It has shown satisfactory results, outperforming the existing systems. It thus testifies to the agenda of quality over quantity.

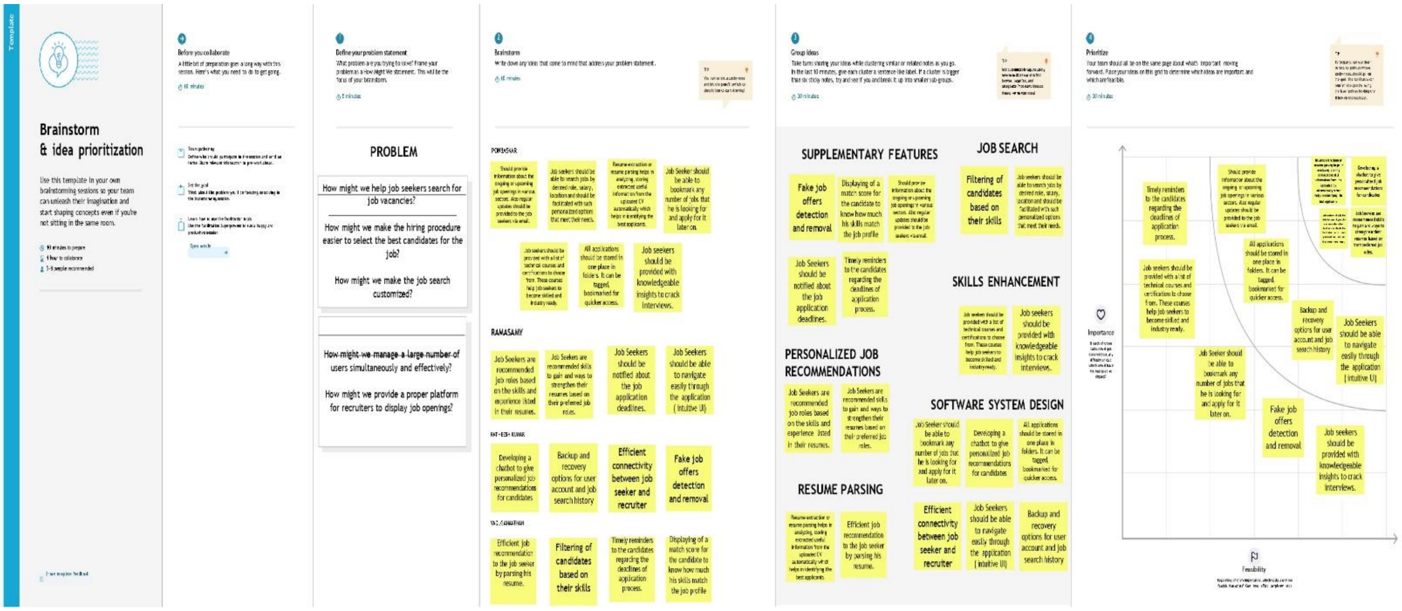
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS


Who are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
Getting ready for exams
Studying abroad
A psychology student
What do they hear?
From friends? From colleagues?
What are they hearing second-hand?
It's important to have a network
What do they think and feel?
What are their fears, frustrations, and anxieties?
High fees
Grade inflation
Expensive study material
What are their wants, needs, hopes, and dreams?
Wants to help people
What other thoughts and feelings might influence their behavior?
What do they need to do?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How do we know they're successful?
Pass the final exams
Make friends
What do they see?
In their environment?
In the marketplace?
What do they see others say and do?
What do they watch & read?
Video calls
Course literature
Blogs
What do they say?
What have we heard them say? What can we magine them saying?
What do they do?
What behavior have we observed?
What can we imagine them doing?
Attends every class
Talks to a counsellor for extra help

3.2 IDEATION & BRAINSTORMING



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

3-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering

Set the goal

Learn how to use the facilitation tools

Open article

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we help job seekers search for job vacancies?

How might we make the hiring procedure easier to select the best candidates for the job?

How might we make the job search customized?

~~How might we manage a large number of~~ users simultaneously and effectively?

How might we provide a proper platform for recruiters to display job openings?

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

POMBASKAR

- Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
- Job seekers should be able to search jobs by desired role, salary, location and should be facilitated with such personalized options that meet their needs.
- Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants.
- Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
- Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
- All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
- Job seekers should be provided with knowledgeable insights to crack interviews.

RAMASAMY

- Job Seekers are recommended job roles based on the skills and experience listed in their resumes.
- Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.
- Job Seekers should be notified about the job application deadlines.
- Job Seekers should be able to navigate easily through the application (intuitive UI)

RAT-EESH KUMAR

- Developing a chatbot to give personalized job recommendations for candidates
- Backup and recovery options for user account and job search history
- Efficient connectivity between job seeker and recruiter
- Fake job offers detection and removal

VAS.ANATHAN

- Efficient job recommendation to the job seeker by parsing his resume.
- Filtering of candidates based on their skills
- Timely reminders to the candidates regarding the deadlines of application process.
- Displaying of a match score for the candidate to know how much his skills match the job profile

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

SUPPLEMENTARY FEATURES

- Fake job offers detection and removal
- Displaying of a match score for the candidate to know how much his skills match the job profile
- Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
- Job Seekers should be notified about the job application deadlines.
- Timely reminders to the candidates regarding the deadlines of application process.

JOB SEARCH

- Filtering of candidates based on their skills
- Job seekers should be able to search jobs by desired role, salary, location and should be facilitated with such personalized options that meet their needs.

SKILLS ENHANCEMENT

- Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
- Job seekers should be provided with knowledgeable insights to crack interviews.

PERSONALIZED JOB RECOMMENDATIONS

- Job Seekers are recommended job roles based on the skills and experience listed in their resumes.
- Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.

SOFTWARE SYSTEM DESIGN

- Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
- Developing a chatbot to give personalized job recommendations for candidates
- All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
- Efficient connectivity between job seeker and recruiter
- Job Seekers should be able to navigate easily through the application (intuitive UI)
- Backup and recovery options for user account and job search history

RESUME PARSING

- Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants
- Efficient job recommendation to the job seeker by parsing his resume.

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

- Timely reminders to the candidates regarding the deadlines of application process.
- Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
- Developing a chatbot to give personalized job recommendations for candidates
- Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.
- Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
- All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
- Backup and recovery options for user account and job search history
- Job Seekers should be able to navigate easily through the application (intuitive UI)
- Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
- Fake job offers detection and removal
- Job seekers should be provided with knowledgeable insights to crack interviews.

3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement	Nowadays a lot of students have great skills but unable to get a desired/appropriate job, so an end-to-end web application can be created which is capable of displaying current job openings based on user skill set making it easier to hire and get hired.
2	Idea / Solution Description	To develop an end-to-end web application which in default have a lot of current job openings through job search API out of which appropriate job will be recommended based on user skill set. At the same time students can develop their skills side by side with various courses and webinars offered by reputed organization. In addition to this a smart chat bot will be available for 24*7 which can help users in finding the right job.

3	Novelty / Uniqueness	To develop an end-to-end web application which in default have a lot of current job openings through job search API out of which appropriate job will be recommended based on user skill set. At the same time students can develop their skills side by side with various courses and webinars offered by reputed organization. In addition to this a smart chat bot will be available for 24*7 which can help users in finding the right job.
4	Social Impact / Customer Satisfaction	Students will be benefited as they will get to know which job suits them based on theirskill set and therefore Lack of Unemployment can be reduced
5	Business Model	Students will be benefited as they will get to know which job suits them based on theirskill set and therefore Lack of Unemployment can be reduced
6	Scalability of the Solution	The application can easily be scaled to accept multiple inputs and process them parallelly to further increase efficiency

3.4 PROBLEM SOLUTION FIT

CUSTOMER SEGMENT:

1. Students who are looking forward for internships to improvise their skills.
2. Freshers who have no experience but have skills and a seeking for a job
3. Experienced people who are looking forward to upgrade them professionally
4. Professionals who are expecting work from home
5. Technical and non-technical job seekers

JOBS-TO-BE-DONE & PROBLEMS:

1. People want to know about all the job openings at their own pace
2. Need for a one stop destination where all kind of jobs can be found
3. Alert mechanism to not miss any appropriate job openings

TRIGGER:

People want a one stop destination where they can find all the job listings available

AVAILABLE SOLUTION:

1. Breezy is a cloud based recruiting and applicant tracking platform for small and mid-size businesses
2. Bootstrap is used to create a branded career sight and distribute listings to over 50 job boards

CUSTOMER CONSTRAINT:

1. To visit on-site each time in search of job
2. Need to search different website each time they need to apply a job
3. Manually filtering the job based on their skill-set
4. Need to find the relevant job

PROBLEM ROOT CAUSE

1. They need to visit each and every company in person every time
2. Online websites available are specific for each company and consumes time
3. Other applications available are complex for the user to handle
4. Missing valuable opportunities due to lack of time management.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR.NO	FUNCTIONAL REQUIREMENTS	SUB REQUIREMENTS
FR-1	**User Registration**	The job seeker Register her /him application form through form hardcopy (or) Register through Gmail (or) Register through Linkedin.
FR-2	**User Confirmation**	Job seeker find application Confirmation via Email (or) via OTP.
FR-3	**User Status**	Job seeker find their current status through view dashboard (login with ID and Password).

4.2 NON FUNCTIONAL REQUIREMENTS

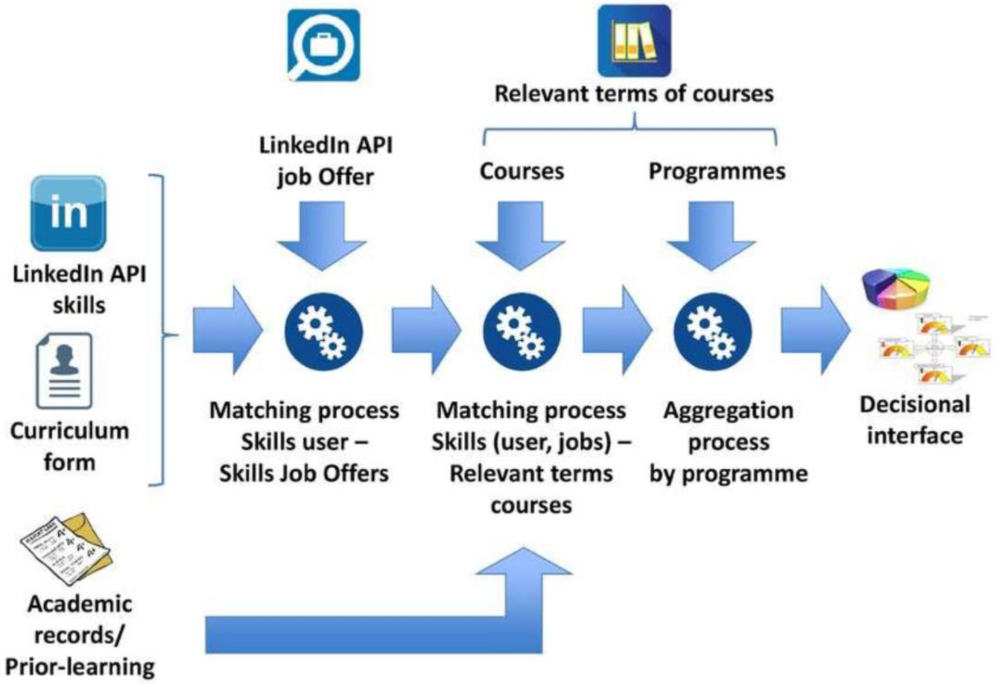
NFR	NON-FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	The error rate of users submitting their payment details at the checkout page mustn't exceed 10 percent.
NFR-2	Security	If your security relies on specific standards and encryption methods, these

		standards don’t directly describe the behavior of a system, but rather help engineers with implementation guides
NFR-3	Reliability	He system must perform without failure in 95 percent of use cases during a month.
NFR-4	Performance	The landing page supporting 5,000 users per hour must provide 6 second or less response time in a Chrome desktop browser, including the rendering of text and images and over an LTE connection
NFR-5	Availability	The web dashboard must be available to US users 99.98 percent of the time every month during business hours EST.
NFR-6	Scalability	The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance

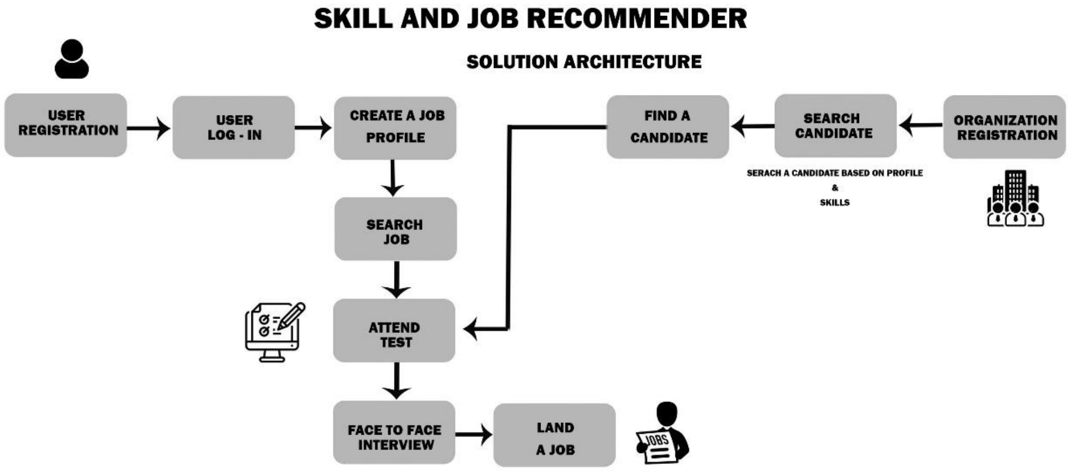
CHAPTER 5

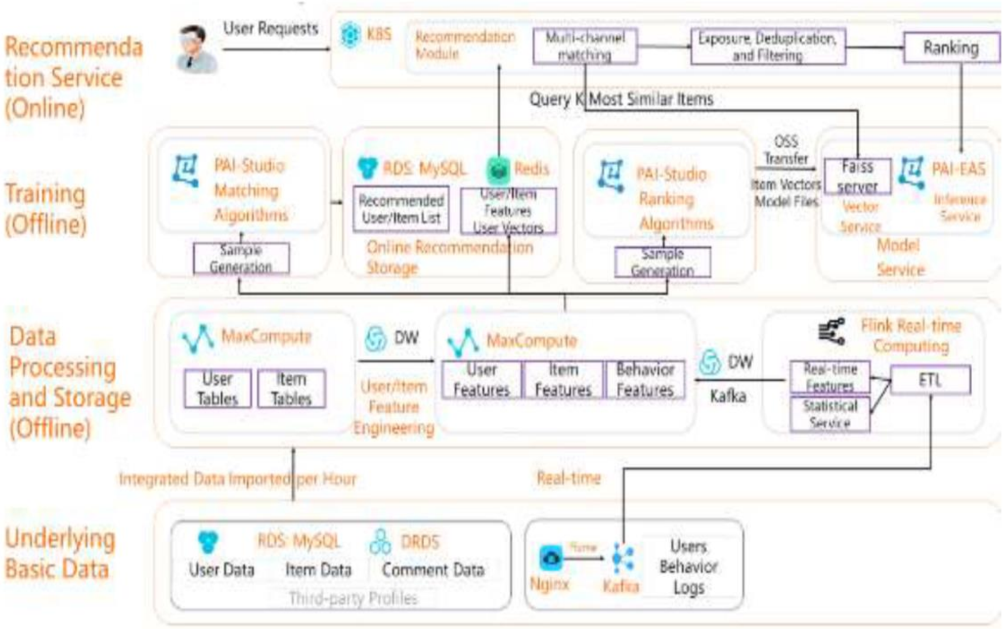
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM


Relevant terms of courses
LinkedIn API job Offer
Courses
Programmes
LinkedIn API skills
Curriculum form
Matching process Skills user – Skills Job Offers
Matching process Skills (user, jobs) – Relevant terms courses
Aggregation process by programme
Decisional interface
Academic records/ Prior-learning

5.2 SOLUTION & TECHNICAL ARCHITECTURE




Recommendation Service (Online)
User Requests
K8S
Recommendation Module
Multi-channel matching
Exposure, Deduplication, and Filtering
Ranking
Query K Most Similar Items
Training (Offline)
PAI-Studio Matching Algorithms
Sample Generation
RDS: MySQL
Redis
Recommended User/Item List
User/Item Features User Vectors
Online Recommendation Storage
PAI-Studio Ranking Algorithms
Sample Generation
OSS Transfer
Item Vectors Model Files
Faiss server
Vector Service
PAI-EAS
Inference Service
Model Service
Data Processing and Storage (Offline)
MaxCompute
User Tables
Item Tables
DW
User/Item Feature Engineering
MaxCompute
User Features
Item Features
Behavior Features
DW
Kafka
Flink Real-time Computing
Real-time Features
Statistical Service
ETL
Integrated Data Imported per Hour
Real-time
Underlying Basic Data
RDS: MySQL
DRDS
User Data
Item Data
Comment Data
Third-party Profiles
Nginx
Flume
Kafka
Users Behavior Logs

5.3 USER STORIES

User Type	Functional Requirements	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Accessing the Application	USN-1	As a user, I can register for the application by entering my email, password and confirming my password	I can access my account dashboard	High	5-10 mins
	Confirmation Mail	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	6-7mins
	Viewing the Results	USN-3	As a user, I can register for the application through Online	Can see and access my application through Login	Low	15-20 mins

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account /dashboard	High	Annamalai
Sprint-1		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive confirmation email & click confirm	High	Karthick Raja
Sprint-2		USN-3	As a user, I can register for the applicationthrough Facebook	I can register & access the dashboard with FacebookLogin	Low	Kishore
Sprint-1		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Yogesh
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Annamalai
Sprint-1	Dashboard	USN-6	Create a model set that contains those models, then assign it to arole.	Assign that group to theappropriate roles on the Roles page	High	Karthick Raja
Sprint-1	Identity-Aware	USN-7	Open, public access, User-authenticated access, Employee-restricted access.	Company public website. App runningon the company intranet. App with access to customer private information.	High	Yogesh

Sprint-1	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how to tackle customer queries.	Medium	Kishore
Sprint-1	Device management	USN-9	You can Delete/Disable/Enable devicesin Azure Active Directory butyou cannot Add/Remove Users in the directory.	Ease of use.	Medium	Annamalai

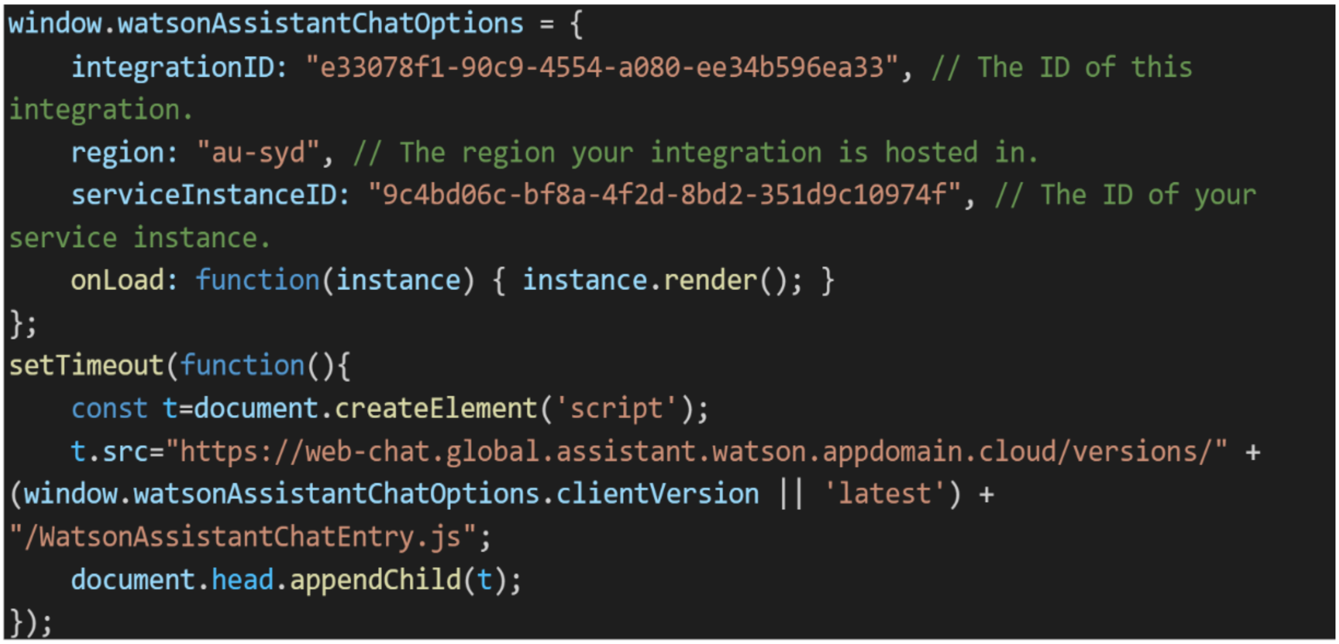
6.2 SPRINT DELIVERY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint - I	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint - II	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint - III	20	6 Days	07 Oct 2022	12 Nov 2022	20	12 Nov 2022
Sprint - IV	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

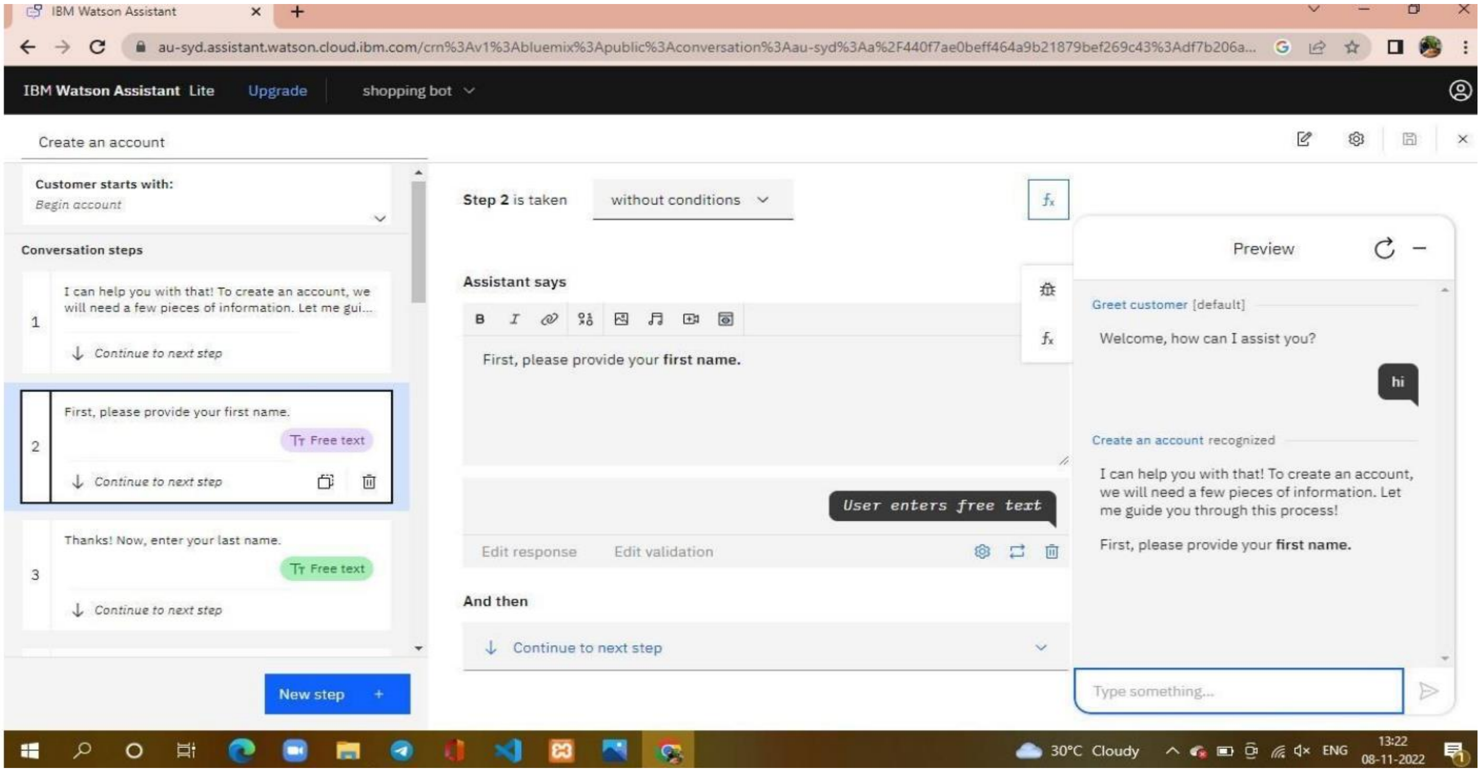
The software has an In-built “Chat Bot” which can help assist with ongoing queries and provide fast and effective solutions to user problems which may occur and also redirect to PNT2022TMID20801 management attention if need be there any complications the customer service will be available 24*7 to assist in case of any controversial issues arise.



```
window.watsonAssistantChatOptions = {
    integrationID: "e33078f1-90c9-4554-a080-ee34b596ea33", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "9c4bd06c-bf8a-4f2d-8bd2-351d9c10974f", // 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);
});
```

In this project we have created the dashboard page to view the jobs available and to make ease to access the website

- They communicate information quickly.
- They display information clearly and efficiently.
- They show trends and changes in data over time.
- They are easily customizable.
- The most important widgets and data components are effectively presented in a limited space.



1 I can help you with that! To create an account, we will need a few pieces of information. Let me gui...

2 First, please provide your first name.

3 Thanks! Now, enter your last name.

First, please provide your **first name.**

Greet customer [default]

Welcome, how can I assist you?

hi

Create an account recognized

I can help you with that! To create an account, we will need a few pieces of information. Let me guide you through this process!

First, please provide your **first name.**

CHAPTER 8

TESTING

Test Cases

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.

Defect Analysis

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

Resolution	Severity1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	2	18
Duplicate	1	1	3	1	6
External	2	3	0	1	6
Fixed	11	2	4	20	37
Skipped	0	0	1	1	2
Won’t Fix	0	5	2	1	8
Totals	24	15	12	26	77

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
PrintEngine	7	0	1	6
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	1	2
Exception Reporting	9	0	1	8
Final Report Output	4	0	1	3
Version Control	2	0	0	2

6.3 User Acceptance Testing

Purpose of Document The purpose of this document is to briefly explain the test coverage and open issues of the Inventory Management System project at the time of the release to User Acceptance Testing (UAT) User Acceptance Testing is carried out in a separate testing environment. A change, an update, or a new feature is requested and developed. Unit and integration tests are run. All seems to be in order. But then, after it is released to the public, serious problems appear. Rework and retesting are not the most expensive consequences when that happens. Loss of reputation is.

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

Based on the two types of user recommendations mentioned above, we analyze the performance of all the techniques mentioned above. The resultant jobs recommended to each new user are then checked with the job that the user is originally in as per the test dataset. If the original user job is recommended in the model result, then the model appends 1 for yes else, it appends 0 for no. This array of 0's and 1's thus received is then checked for accuracy by computing the count of 1's from the total user predictions Among all the models made with the incorporation of different similarity metrics, the cosine similarity based job recommendation system model outperformed rest of them all. The metrics used to analyze the model performance are: accuracy, precision, recall and F1-score.This are because cosine considers the existence of duplicate terms while computing similarity. Also, computationally, cosine has low complexity and ease over handling spare data vectors since only non-zero dimensions are considered. Upon analyzing the result table we observe that the short-comings of some similarity measures upon recommending top 5 and highest-score based job recommendations as even upon achieving high. Similarity scores are due to the fact that users are seen to have different jobs than the ones recommended by the models, thus resulting in 6-10% error rates.

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- The model doesn't need any data about other users, since the recommendations are specific to this user.
- This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

DISADVANTAGES

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
- The model can only make recommendations based on existing interests of the user.
- In other words, the model has limited ability to expand on the users' existing interests

CHAPTER 11

CONCLUSION

In this project, Content-Based Filtering and Collaborative Filtering of recommendations have been compared. Additionally, an aggregation plus recommender system has been devised. Content-Based Filtering recommends the results based on matching the personal preferences of the user with the given document whereas collaborative filtering recommends based on the preferences of fellow users. On evaluating both of these methods, it was concluded that a hybrid system of both of these overcomes the limitations of both of them and increases the efficiency of ranking. Problems of cold start, sparse database, scalability, and lack of trend recommendation have been eliminated. The proposal is to design a Job Recommender system that prioritizes quality over quantity. While there are websites and job listing portals already recommending jobs to job seekers based on their profiles, this research on aggregate quality recommendations has been achieved by crawling selectively, overcoming the limitations. A fully functioning user interface was developed to combine everything together to give the user a seamless experience.

CHAPTER 12

FUTURE SCOPE

Future works in the case of Personalized Job Recommendation Systems are the utilization of the user-preferred location to get job recommendations based on jobs in organizations established in nearby areas.

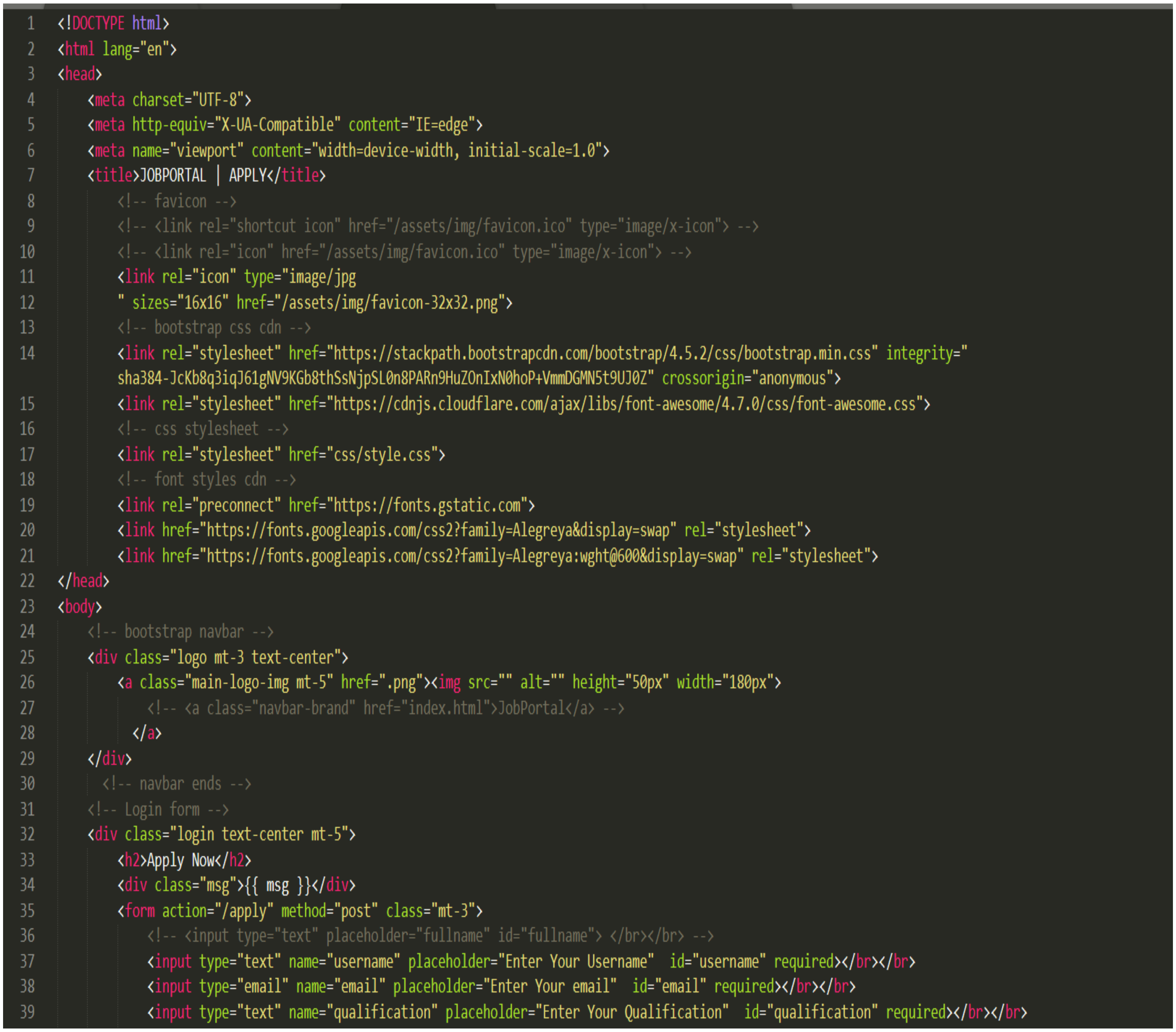
This can be done by extracting the latitudes and longitudes of the user-preferred location and computing the Euclidean distances between the latitudes and longitudes of the organization location. This filters out other jobs that fall far from the user-preferred location and gives a more accurate job recommender as part of the future work, we plan to use features of similar candidates and jobs in sequence information.

As of now, recommendation using similar candidates and jobs forms part of non-machine learning based recommendations and the initial result seem promising. Finally, it would be interesting text end our methodology to other recommender systems.

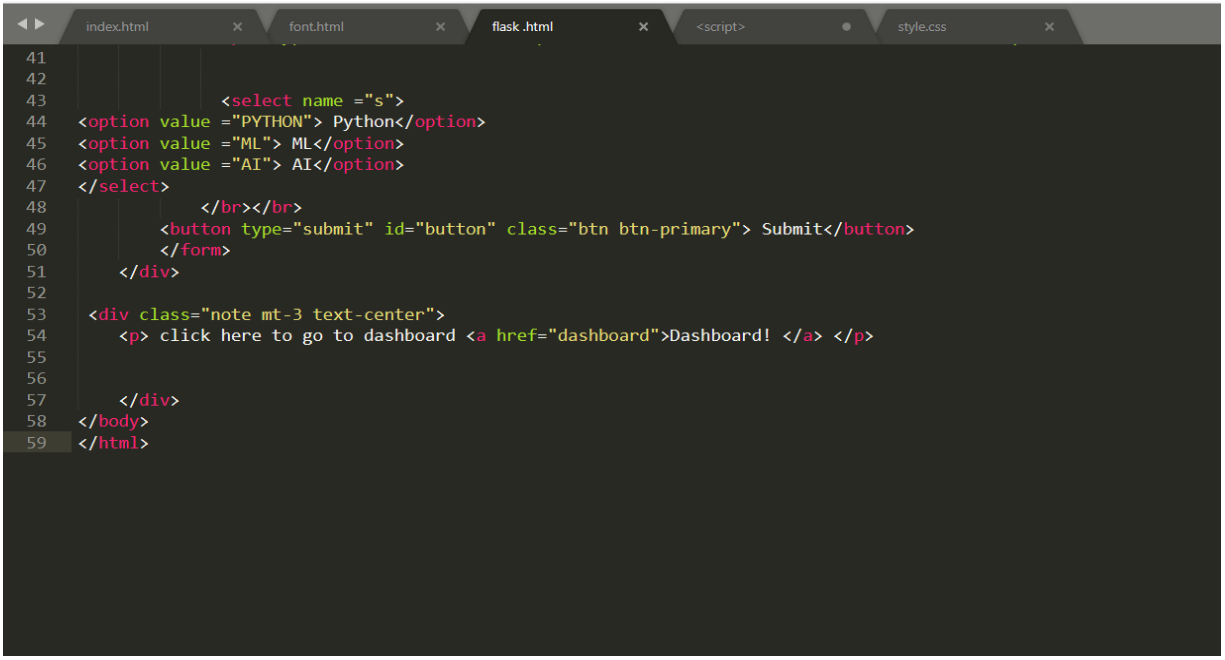
APPENDIX

SOURCE CODE

FLASK APP

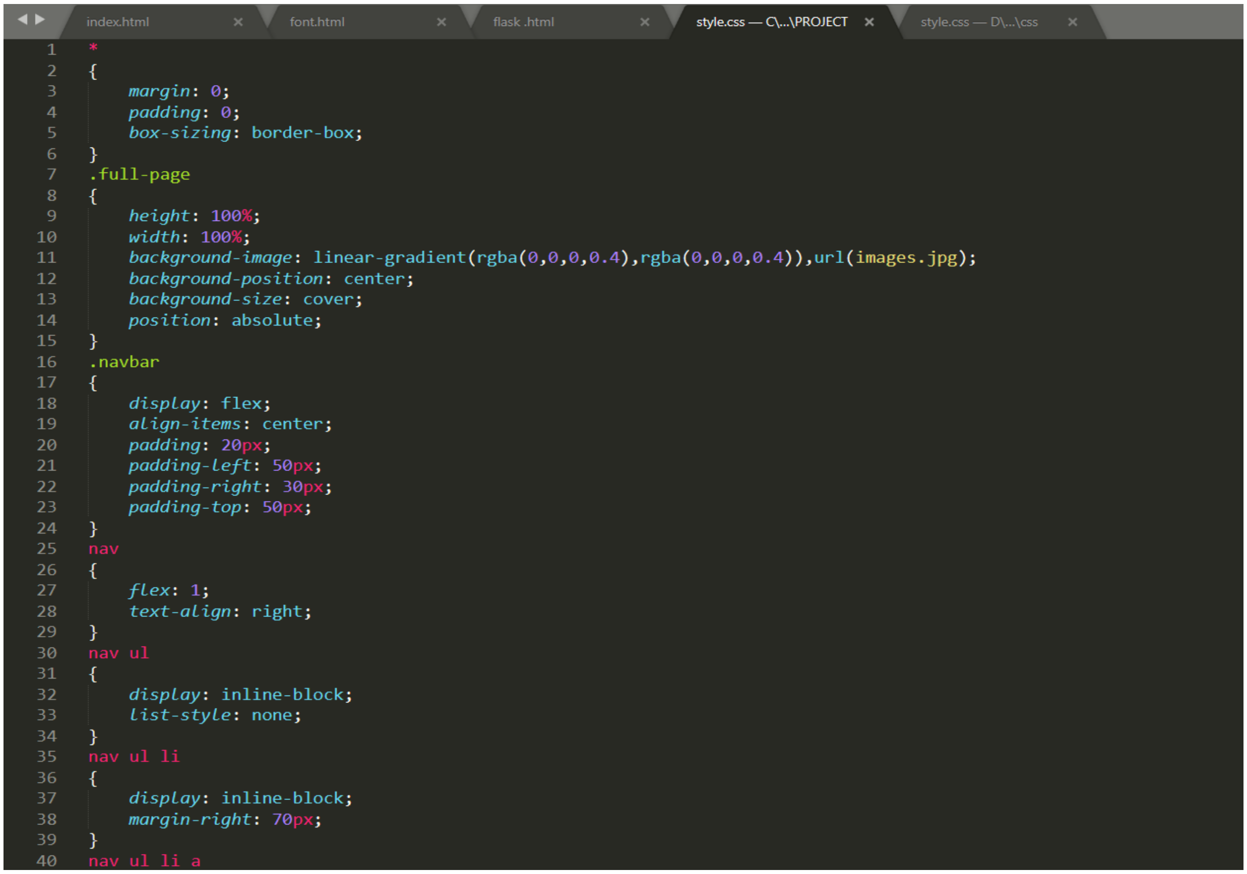


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JOBPORTAL | APPLY</title>
        <!-- favicon -->
        <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
        <!-- <link rel="icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
        <link rel="icon" type="image/jpg
        " sizes="16x16" href="/assets/img/favicon-32x32.png">
        <!-- bootstrap css cdn -->
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="
        sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.css">
        <!-- css stylesheet -->
        <link rel="stylesheet" href="css/style.css">
        <!-- font styles cdn -->
        <link rel="preconnect" href="https://fonts.gstatic.com">
        <link href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap" rel="stylesheet">
        <link href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap" rel="stylesheet">
</head>
<body>
    <!-- bootstrap navbar -->
    <div class="logo mt-3 text-center">
        <a class="main-logo-img mt-5" href=".png"><img src="" alt="" height="50px" width="180px">
            <!-- <a class="navbar-brand" href="index.html">JobPortal</a> -->
          </a>
    </div>
      <!-- navbar ends -->
    <!-- Login form -->
    <div class="login text-center mt-5">
        <h2>Apply Now</h2>
        <div class="msg">{{ msg }}</div>
        <form action="/apply" method="post" class="mt-3">
            <!-- <input type="text" placeholder="fullname" id="fullname"> </br></br> -->
            <input type="text" name="username" placeholder="Enter Your Username"  id="username" required></br></br>
            <input type="email" name="email" placeholder="Enter Your email"  id="email" required></br></br>
            <input type="text" name="qualification" placeholder="Enter Your Qualification"  id="qualification" required></br></br>
```

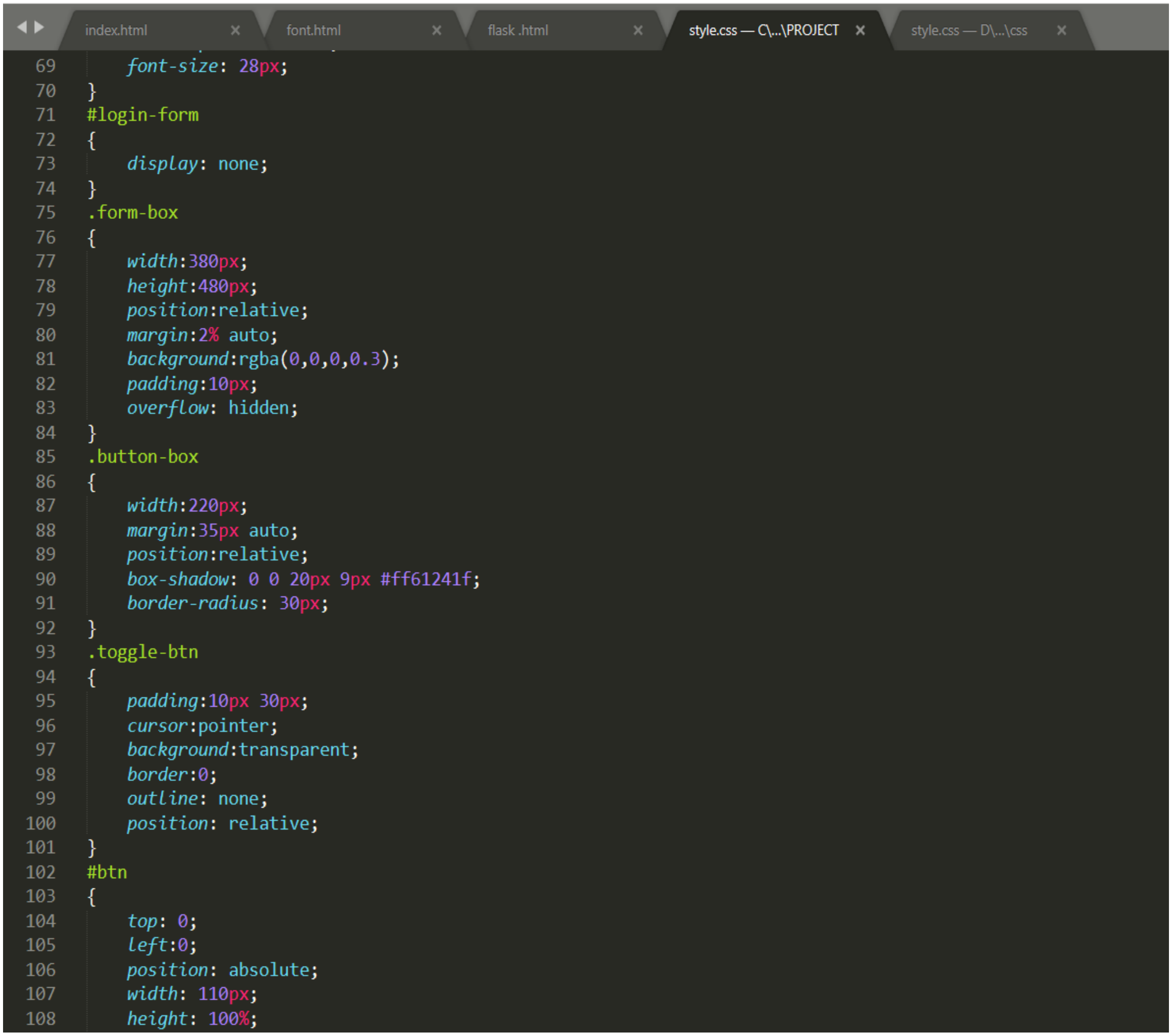


```html


              <select name ="s">
<option value ="PYTHON"> Python</option>
<option value ="ML"> ML</option>
<option value ="AI"> AI</option>
</select>
            </br></br>
        <button type="submit" id="button" class="btn btn-primary"> Submit</button>
        </form>
    </div>

 <div class="note mt-3 text-center">
    <p> click here to go to dashboard <a href="dashboard">Dashboard! </a> </p>


    </div>
</body>
</html>
```

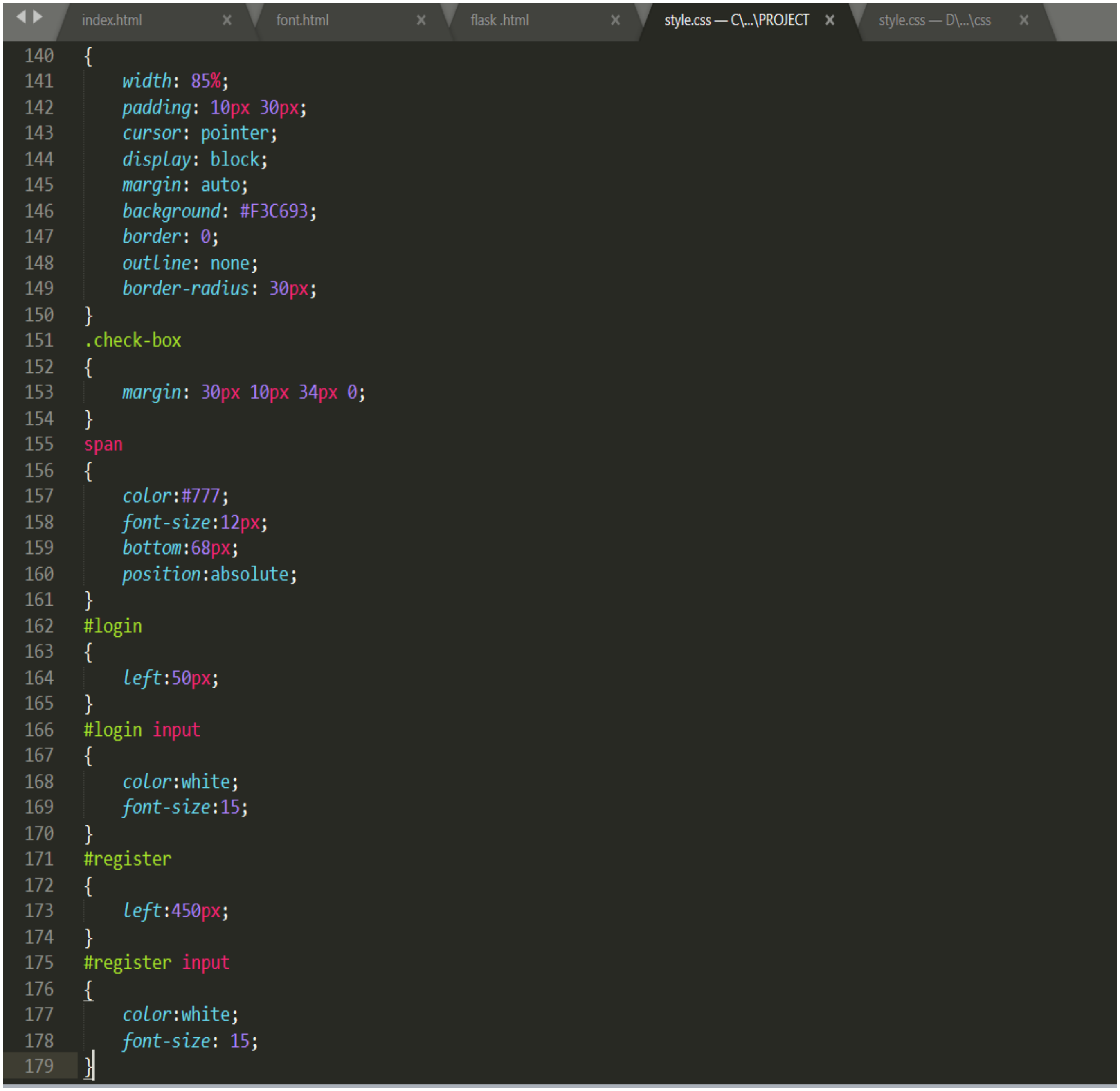
Css



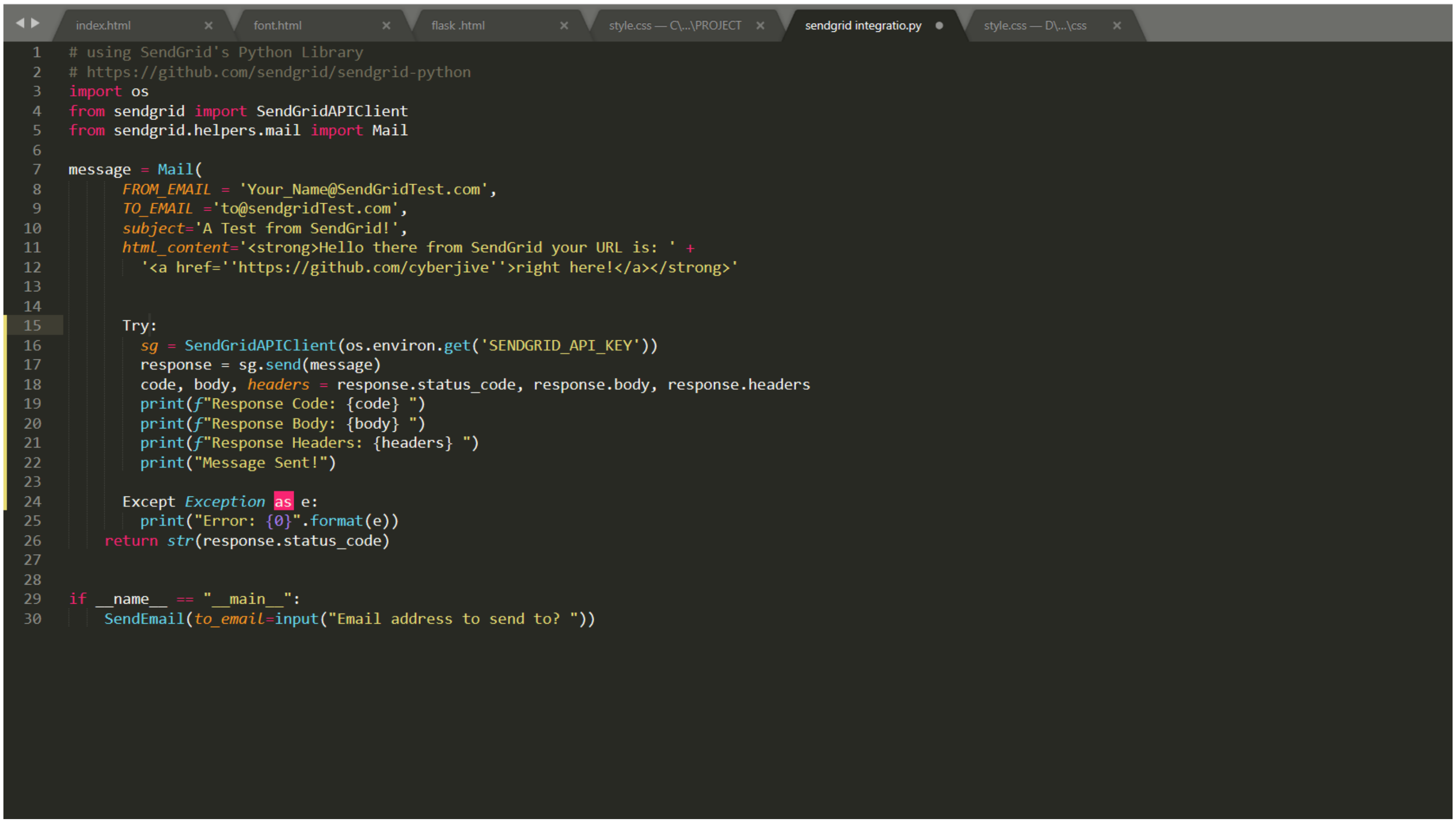
```css
*
{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
.full-page
{
    height: 100%;
    width: 100%;
    background-image: linear-gradient(rgba(0,0,0,0.4),rgba(0,0,0,0.4)),url(images.jpg);
    background-position: center;
    background-size: cover;
    position: absolute;
}
.navbar
{
    display: flex;
    align-items: center;
    padding: 20px;
    padding-left: 50px;
    padding-right: 30px;
    padding-top: 50px;
}
nav
{
    flex: 1;
    text-align: right;
}
nav ul
{
    display: inline-block;
    list-style: none;
}
nav ul li
{
    display: inline-block;
    margin-right: 70px;
}
nav ul li a
```



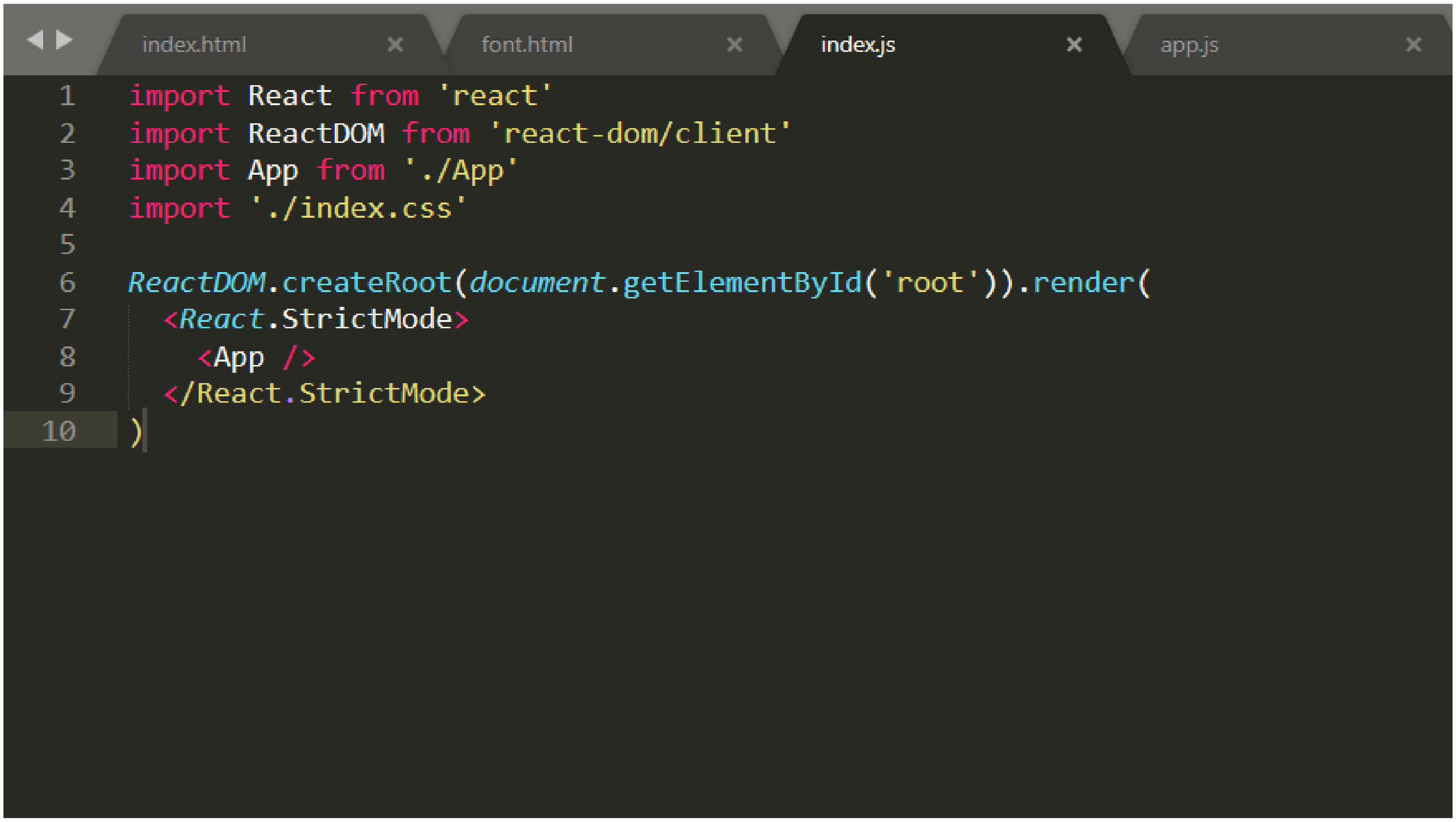
```
    font-size: 28px;
}
#login-form
{
    display: none;
}
.form-box
{
    width:380px;
    height:480px;
    position:relative;
    margin:2% auto;
    background:rgba(0,0,0,0.3);
    padding:10px;
    overflow: hidden;
}
.button-box
{
    width:220px;
    margin:35px auto;
    position:relative;
    box-shadow: 0 0 20px 9px #ff61241f;
    border-radius: 30px;
}
.toggle-btn
{
    padding:10px 30px;
    cursor:pointer;
    background:transparent;
    border:0;
    outline: none;
    position: relative;
}
#btn
{
    top: 0;
    left:0;
    position: absolute;
    width: 110px;
    height: 100%;
```



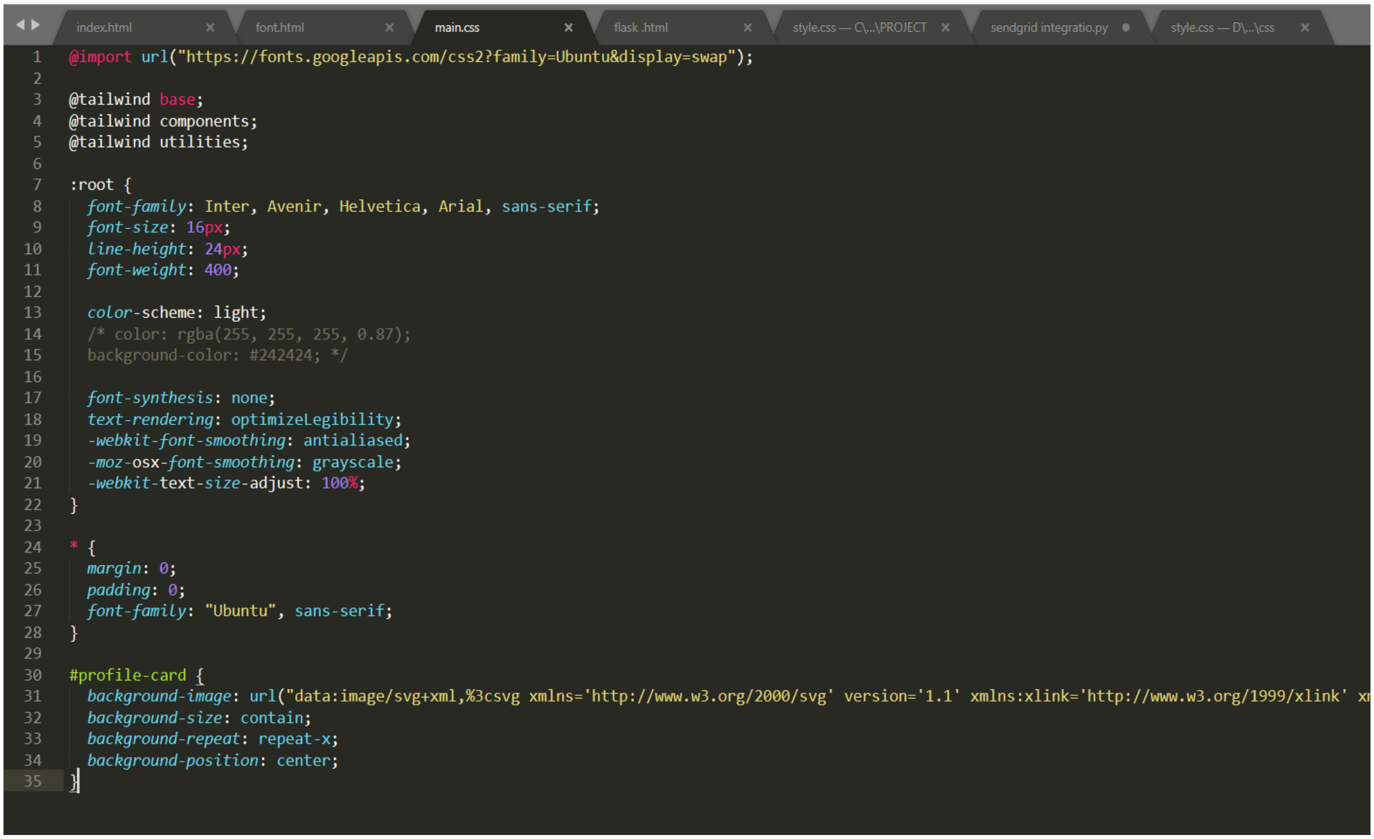
```
{
    width: 85%;
    padding: 10px 30px;
    cursor: pointer;
    display: block;
    margin: auto;
    background: #F3C693;
    border: 0;
    outline: none;
    border-radius: 30px;
}
.check-box
{
    margin: 30px 10px 34px 0;
}
span
{
    color:#777;
    font-size:12px;
    bottom:68px;
    position:absolute;
}
#login
{
    left:50px;
}
#login input
{
    color:white;
    font-size:15;
}
#register
{
    left:450px;
}
#register input
{
    color:white;
    font-size: 15;
}
```



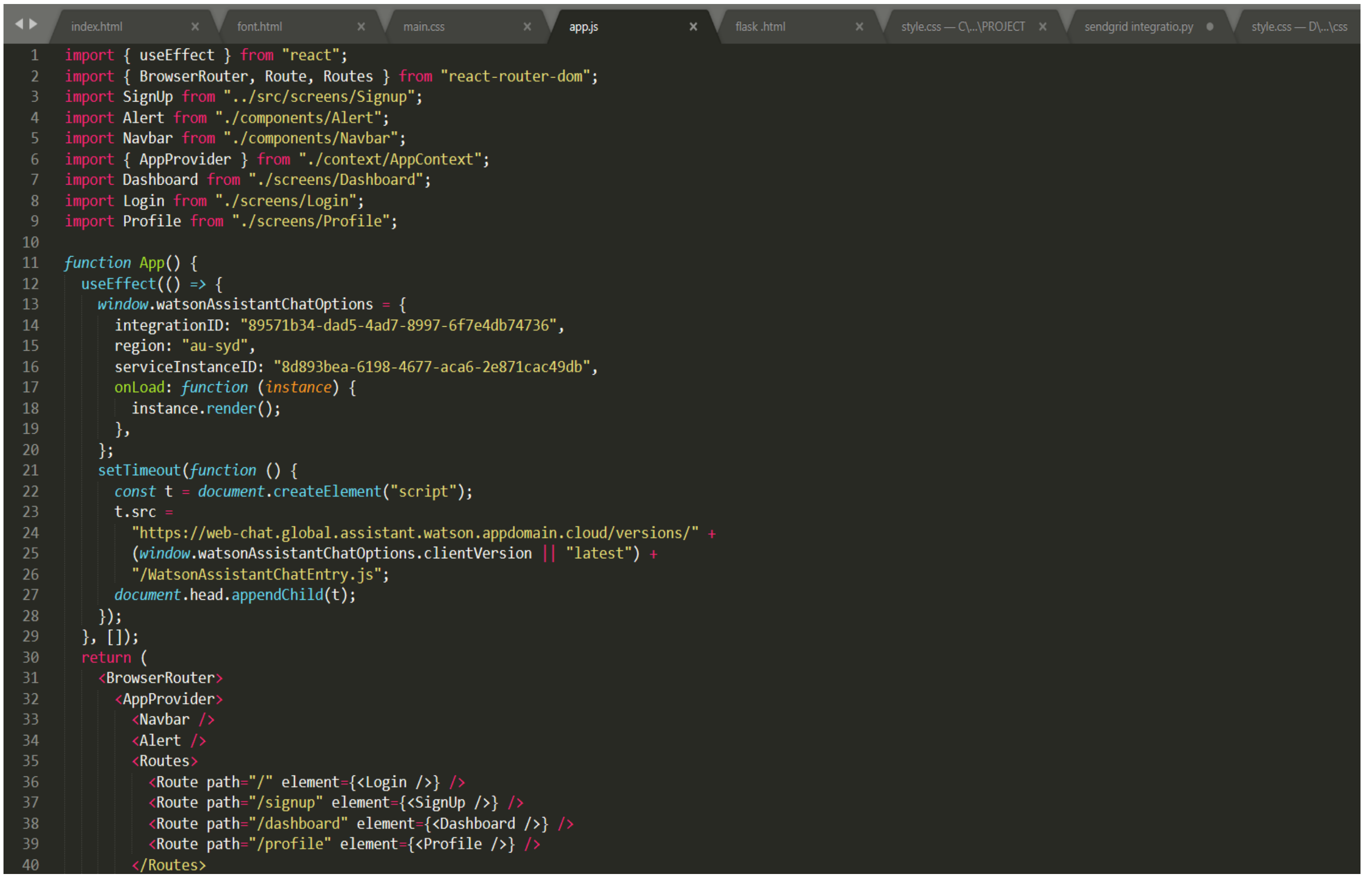
```python
# using SendGrid's Python Library
# https://github.com/sendgrid/sendgrid-python
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

message = Mail(
      FROM_EMAIL = 'Your_Name@SendGridTest.com',
      TO_EMAIL ='to@sendgridTest.com',
      subject='A Test from SendGrid!',
      html_content='<strong>Hello there from SendGrid your URL is: ' +
        '<a href=''https://github.com/cyberjive''>right here!</a></strong>'


      Try:
        sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
        response = sg.send(message)
        code, body, headers = response.status_code, response.body, response.headers
        print(f"Response Code: {code} ")
        print(f"Response Body: {body} ")
        print(f"Response Headers: {headers} ")
        print("Message Sent!")

      Except Exception as e:
        print("Error: {0}".format(e))
    return str(response.status_code)


if __name__ == "__main__":
    SendEmail(to_email=input("Email address to send to? "))
```



```javascript
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
)
```



```css
@import url("https://fonts.googleapis.com/css2?family=Ubuntu&display=swap");

@tailwind base;
@tailwind components;
@tailwind utilities;

:root {
  font-family: Inter, Avenir, Helvetica, Arial, sans-serif;
  font-size: 16px;
  line-height: 24px;
  font-weight: 400;

  color-scheme: light;
  /* color: rgba(255, 255, 255, 0.87);
  background-color: #242424; */

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  -webkit-text-size-adjust: 100%;
}

* {
  margin: 0;
  padding: 0;
  font-family: "Ubuntu", sans-serif;
}

#profile-card {
  background-image: url("data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' version='1.1' xmlns:xlink='http://www.w3.org/1999/xlink' x
  background-size: contain;
  background-repeat: repeat-x;
  background-position: center;
}
```



```javascript
import { useEffect } from "react";
import { BrowserRouter, Route, Routes } from "react-router-dom";
import SignUp from "../src/screens/Signup";
import Alert from "./components/Alert";
import Navbar from "./components/Navbar";
import { AppProvider } from "./context/AppContext";
import Dashboard from "./screens/Dashboard";
import Login from "./screens/Login";
import Profile from "./screens/Profile";

function App() {
  useEffect(() => {
    window.watsonAssistantChatOptions = {
      integrationID: "89571b34-dad5-4ad7-8997-6f7e4db74736",
      region: "au-syd",
      serviceInstanceID: "8d893bea-6198-4677-aca6-2e871cac49db",
      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);
    });
  }, []);
  return (
    <BrowserRouter>
      <AppProvider>
        <Navbar />
        <Alert />
        <Routes>
          <Route path="/" element={<Login />} />
          <Route path="/signup" element={<SignUp />} />
          <Route path="/dashboard" element={<Dashboard />} />
          <Route path="/profile" element={<Profile />} />
        </Routes>
```

PROJECT DEMONSTARTION VIDEO UPLOADED HERE

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-19105-1659693271

PROJECT DEMO LINK :

https://drive.google.com/file/d/1msxYYj1dP8cNWwepgJn7EaNQGFGfCB6R/view?usp=share_link