TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES – MANILA
College of Computer Studies

Bachelor of Science in Computer Science
S.Y. 2023-2024, Summer Term
Software Engineering 2
CCS 308

Continuous Development Task for Software Engineering 2

SUBMITTED BY
Group 3 – CS33S2

Members:

Aguilar, Cristelle Andrea

Cofreros, Patrick Jay

Pangilinan, Jayalle

DATE

June 28, 2024

SUBMITTED TO

Sir. Mar Eli Sagsagat

Development Contribution

Aguilar, Cristelle Andrea

```
# Main function to run the application
def run():
    st.title("SkillSync: Job Recommendation")
    choice = "Normal User"

    # Handling database and tables
    connection = sqlite3.connect("se2project.db")

    # connection = pymysql.connect(host='localhost', user='root', password='')
    cursor = connection.cursor()
    # cursor.execute("CREATE DATABASE IF NOT EXISTS SRA;")
    # connection.select_db("SRA")
    table_sql = """
                CREATE TABLE IF NOT EXISTS user_data (
                    ID INTEGER PRIMARY KEY AUTOINCREMENT,
                    Name VARCHAR(100) NOT NULL,
                    Email_ID VARCHAR(50) NOT NULL,
                    Resume_Score VARCHAR(8) NOT NULL,
                    Timestamp VARCHAR(50) NOT NULL,
                    Page_no VARCHAR(5) NOT NULL,
                    Predicted_Field VARCHAR(25) NOT NULL,
                    User_level VARCHAR(30) NOT NULL,
                    Actual_skills VARCHAR(300) NOT NULL,
                    Recommended_skills VARCHAR(300) NOT NULL,
                    Recommended_courses VARCHAR(600) NOT NULL
                );
                """
```

```
    # Normal User functionality
    if choice == 'Normal User':
        pdf_file = st.file_uploader("Choose your Resume", type=["pdf"])
        if pdf_file is not None:
            save_image_path = './Uploaded_Resumes/' + pdf_file.name
            with open(save_image_path, "wb") as f:
                f.write(pdf_file.getbuffer())
            show_pdf(save_image_path)
            resume_data = ResumeParser(save_image_path).get_extracted_data()

            if resume_data:
                st.header("**Resume Analysis**")
                st.success("Hello " + resume_data['name'])
                st.subheader("**Your Basic info**")
                try:
                    st.text('Name: ' + resume_data['name'])
                    st.text('Email: ' + resume_data['email'])
                    st.text('Contact: ' + resume_data['mobile_number'])
                    st.text('Resume pages: ' + str(resume_data['no_of_pages']))
                except KeyError:
                    pass
```

This Python code creates a Streamlit web application titled "SkillSync: Job Recommendation System". It connects to a SQLite database named "se2project.db" and ensures a table named "user_data" exists for storing resume information.

For a "Normal User", the app allows uploading a PDF resume. Once uploaded, it saves the file locally and extracts key details like name, email, and contact from the resume using `ResumeParser`. If successful, it displays the parsed information such as name, email, and resume pages using Streamlit's interface.

The parsed data is then stored in the SQLite database table "user_data", ready for further analysis or recommendations related to job skills and courses.

```
##  recommendation
ds_keyword = ['tensorflow', 'keras', 'pytorch', 'machine learning', 'deep Learning', 'flask',
'streamlit']
web_keyword = ['react', 'django', 'node jS', 'react js', 'php', 'laravel', 'magento', 'wordpress',
'javascript', 'angular js', 'c#', 'flask']
android_keyword = ['android', 'android development', 'flutter', 'kotlin', 'xml', 'kivy']
ios_keyword = ['ios', 'ios development', 'swift', 'cocoa', 'cocoa touch', 'xcode']
uiux_keyword = ['ux', 'adobe xd', 'figma', 'zeplin', 'balsamiq', 'ui', 'prototyping', 'wireframes',
                'storyframes', 'adobe photoshop', 'photoshop', 'editing', 'adobe illustrator',
                'illustrator', 'adobe after effects', 'after effects', 'adobe premier pro',
                'premier pro', 'adobe indesign', 'indesign', 'wireframe', 'solid', 'grasp',
                'user research', 'user experience']

recommended_skills = []
reco_field = ''
rec_course = ''
## Courses recommendation
for i in resume_data['skills']:
    ## Data science recommendation
    if i.lower() in ds_keyword:
        print(i.lower())
        reco_field = 'Data Science'
        st.success("** Our analysis says you are looking for Data Science Jobs.**")
        recommended_skills = ['Data Visualization', 'Predictive Analysis', 'Statistical Modeling',
                              'Data Mining', 'Clustering & Classification', 'Data Analytics',
                              'Quantitative Analysis', 'Web Scraping', 'ML Algorithms', 'Keras',
                              'Pytorch', 'Probability', 'Scikit-learn', 'Tensorflow', "Flask",
                              'Streamlit']
```

```
## Android App Development
elif i.lower() in android_keyword:
    print(i.lower())
    reco_field = 'Android Development'
    st.success("** Our analysis says you are looking for Android App Development Jobs **")
    recommended_skills = ['Android', 'Android development', 'Flutter', 'Kotlin', 'XML', 'Java',
                          'Kivy', 'GIT', 'SDK', 'SQLite']
    break

## IOS App Development
elif i.lower() in ios_keyword:
    print(i.lower())
    reco_field = 'IOS Development'
    st.success("** Our analysis says you are looking for IOS App Development Jobs **")
    recommended_skills = ['IOS', 'IOS Development', 'Swift', 'Cocoa', 'Cocoa Touch', 'Xcode',
                          'Objective-C', 'SQLite', 'Plist', 'StoreKit', "UI-Kit", 'AV Foundation',
                          'Auto-Layout']
    break

## Ui-UX Recommendation
elif i.lower() in uiux_keyword:
    print(i.lower())
    reco_field = 'UI-UX Development'
    st.success("** Our analysis says you are looking for UI-UX Development Jobs **")
    recommended_skills = ['UI', 'User Experience', 'Adobe XD', 'Figma', 'Zeplin', 'Balsamiq',
                          'Prototyping', 'Wireframes', 'Storyframes', 'Adobe Photoshop', 'Editing',
                          'Illustrator', 'After Effects', 'Premier Pro', 'Indesign', 'Wireframe',
                          'Solid', 'Grasp', 'User Research']
    break
```

This code analyzes the skills listed in a user's resume to recommend relevant job fields and additional skills that can enhance their employability. It iterates through the skills extracted from the resume, comparing each skill against predefined keywords associated with specific job fields such as Data Science, Web Development, Android Development, IOS Development, and UI-UX Development. When a match is found, it assigns the corresponding job field to `reco_field` and populates `recommended_skills` with a list of additional skills relevant to that field. Additionally, it displays a success message indicating the inferred job field and suggests skills that could boost the user's chances of getting a job in that field. This system helps users understand which job fields they are best suited for based on their current skills and what additional skills they might consider acquiring.

Cofreros, Patrick Jay T.

```
    skills_df = pd.DataFrame(resume_data["skills"], columns=["SKILLS"])
    st.table(skills_df)

    experience_df = pd.DataFrame(resume_data["experience"], columns=["EXPERIENCES"])
    st.table(experience_df)
except KeyError:
    pass
cand_level = ''
if resume_data['no_of_pages'] == 1:
    cand_level = "Fresher"
    st.markdown('''<h4 style='text-align: left; color: #d73b5c;'>You are looking Fresher.</h4>''',
                unsafe_allow_html=True)
elif resume_data['no_of_pages'] == 2:
    cand_level = "Intermediate"
    st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at intermediate level!</h4>''
                unsafe_allow_html=True)
elif resume_data['no_of_pages'] >= 3:
    cand_level = "Experienced"
    st.markdown('''<h4 style='text-align: left; color: #fba171;'>You are at experience level!''',
                unsafe_allow_html=True)

# Insert data into database
insert_data(resume_data['name'], resume_data['email'], res_score: 'NA', str(datetime.datetime.now()),
            resume_data['no_of_pages'], reco_field: 'NA', cand_level, str(resume_data['skills']),
            recommended_skills: 'NA', courses: 'NA')
```

The first lines of code, skills_df and experience_df will create dataframes that contain all the skills and experiences gathered from parsing the resume. It then will be displayed in a tabular format to present all gathered skills and experience. Next, the system will check the number of pages that the resume will contain and display the “level” of which the job seeker is currently in based on the amount of information included in the resume. Lastly, the gathered information from the resume will be inserted into the sqlite database.

Pangilinan, Jayelle

```
# Ensure NLTK data path includes the correct directory
nltk.data.path.append("../Uploaded_Resumes")

# Download NLTK resources if not already downloaded
nltk.download("stopwords")
nltk.download("punkt")

from pydparser import ResumeParser
from pdfminer.high_level import extract_text
from PIL import Image
import sqlite3
import spacy
```

```python
# Function to generate download link for dataframe
def get_table_download_link(df, filename, text):
    csv = df.to_csv(index=False)
    b64 = base64.b64encode(csv.encode()).decode()
    href = f'<a href="data:file/csv;base64,{b64}" download="{filename}">{text}</a>'
    return href

# Function to read PDF and return text
def pdf_reader(file):
    text = extract_text(file)
    return text
```

```python
# Function to display PDF
def show_pdf(file_path):
    with open(file_path, "rb") as f:
        base64_pdf = base64.b64encode(f.read()).decode('utf-8')
    pdf_display = F'<iframe src="data:application/pdf;base64,{base64_pdf}" width="700" height="1000" type="appli
    st.markdown(pdf_display, unsafe_allow_html=True)

# Function to insert data into MySQL database
def insert_data(name, email, res_score, timestamp, no_of_pages, reco_field, cand_level, skills, recommended_skil
                courses):

    connection = sqlite3.connect("se2project.db")
    cursor = connection.cursor()
    DB_table_name = 'user_data'
    insert_sql = f"INSERT INTO {DB_table_name} VALUES (?,?,?,?,?,?,?,?,?,?,?)"
    rec_values = (None,name, email, str(res_score), timestamp, str(no_of_pages), reco_field, cand_level, skills,
                  recommended_skills, courses)
    cursor.execute(insert_sql, rec_values)
    connection.commit()
    connection.close()
```

These functions serve various purposes in a Streamlit application. The **get_table_download_link** function allows users to download a pandas DataFrame as a CSV file by converting the DataFrame to CSV format, encoding it in base64, and generating a clickable download link with a specified filename and link text. The **pdf_reader** function extracts and returns the text content from a given PDF file. The **show_pdf** function displays a PDF file within the Streamlit app by reading the file, encoding it in base64, and using an HTML iframe to show the PDF content on the app's interface. Lastly, the **insert_data** function handles inserting user data into an SQLite database. It connects to the database, inserts the provided information into a table, and then closes the connection. The data includes the user's name, email, resume score, timestamp, number of pages in the resume, recommended field, candidate level, actual skills, recommended skills, and courses.

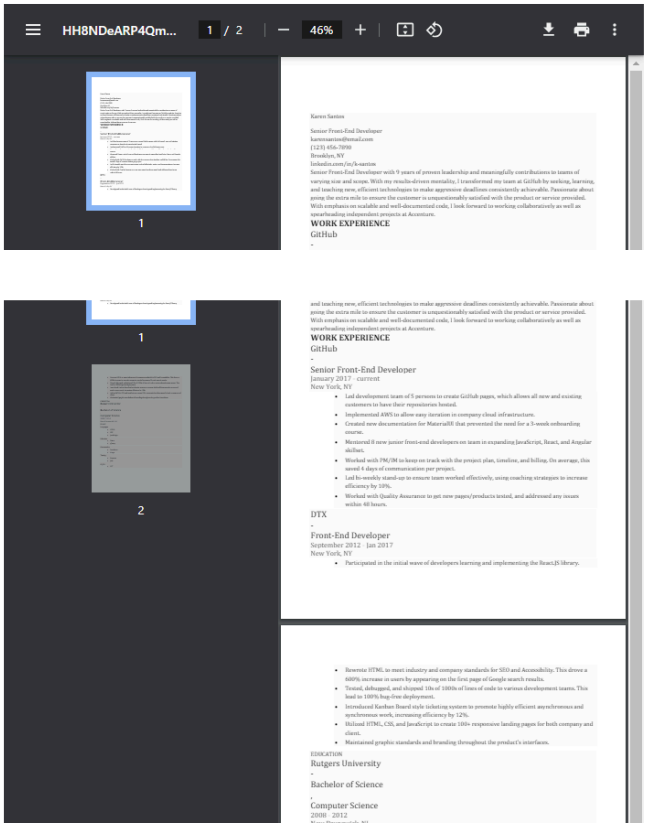
SkillSync: Job Recommendation

Choose your Resume

Drag and drop file here
Limit 200MB per file • PDF

Browse files

UIDEV.pdf 106.1KB



Deploy

Resume Analysis

Hello Karen Santos

Your Basic info

Name: Karen Santos

Email: karensantos@email.com

Contact: (123) 456-7890

Resume pages: 2

	SKILLS
0	Js
1	Cloud
2	Testing
3	Documentation
4	Coaching

4	Coaching
5	Branding
6	Javascript
7	Teaching
8	Quality assurance
9	Css
10	Front-end
11	Seo
12	Billing
13	Aws
14	System
15	Github
16	Communication
17	Plan
18	Vue.js
19	Html

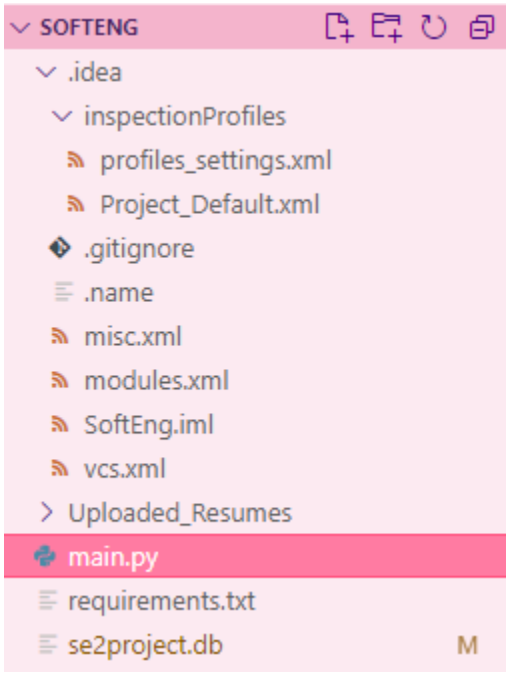
	EXPERIENCES
0	GitHub
1	-
2	Senior Front-End Developer
3	January 2017 - current
4	New York, NY
5	• Led development team of 5 persons to create GitHub pages, which allows all new and existing
6	customers to have their repositories hosted.
7	•
8	Implemented AWS to allow easy iteration in company cloud infrastructure.
9	• Created new documentation for MaterialUI that prevented the need for a 3-week onboarding
10	course.
11	• Mentored 8 new junior front-end developers on team in expanding JavaScript, React, and Angular
12	skillset.
13	• Worked with PM/IM to keep on track with the project plan, timeline, and billing. On average, this
14	saved 4 days of communication per project.
29	•
30	Introduced Kanban Board style ticketing system to promote highly efficient asynchronous and
31	synchronous work, increasing efficiency by 12%.
32	• Utilized HTML, CSS, and JavaScript to create 100+ responsive landing pages for both company and
33	client.
34	• Maintained graphic standards and branding throughout the product's interfaces.

You are at intermediate level!

** Our analysis says you are looking for Web Development Jobs **

Adding this skills to resume will boost 🚀 the chances of getting a Job 💼

Documentation



Syntax

```
import streamlit as st
import pandas as pd
import base64
import datetime
import nltk
import sqlite3
from pydparser import ResumeParser
from pdfminer.high_level import extract_text
from PIL import Image
import spacy

# Ensure NLTK data path includes the correct directory
nltk.data.path.append("../Uploaded_Resumes")

# Download NLTK resources if not already downloaded
nltk.download("stopwords")
nltk.download("punkt")

# Function to read PDF and return text
def pdf_reader(file):
    text = extract_text(file)
```

```
    return text

# Function to display PDF
def show_pdf(file_path):
    with open(file_path, "rb") as f:
        base64_pdf = base64.b64encode(f.read()).decode('utf-8')
    pdf_display = f'<iframe src="data:application/pdf;base64,{base64_pdf}"
width="700" height="1000" type="application/pdf"></iframe>'
    st.markdown(pdf_display, unsafe_allow_html=True)

# Function to insert data into SQLite database
def insert_data(name, email, res_score, timestamp, no_of_pages,
reco_field, cand_level, skills, recommended_skills, courses):
    connection = sqlite3.connect("se2project.db")
    cursor = connection.cursor()
    DB_table_name = 'user_data'
            insert_sql = f"INSERT INTO {DB_table_name} VALUES
(?,?,?,?,?,?,?,?,?,?,?)"
        rec_values = (None, name, email, str(res_score), timestamp,
str(no_of_pages), reco_field, cand_level, skills, recommended_skills,
courses)
    cursor.execute(insert_sql, rec_values)
    connection.commit()
    connection.close()

# Main function to run the application
def run():
    st.title("SkillSync: Job Recommendation")
    choice = "Normal User"

    # Handling database and tables
    connection = sqlite3.connect("se2project.db")
    cursor = connection.cursor()
    table_sql = """
                CREATE TABLE IF NOT EXISTS user_data (
                    ID INTEGER PRIMARY KEY AUTOINCREMENT,
                    Name VARCHAR(100) NOT NULL,
                    Email_ID VARCHAR(50) NOT NULL,
                    Resume_Score VARCHAR(8) NOT NULL,
                    Timestamp VARCHAR(50) NOT NULL,
```

```
                    Page_no VARCHAR(5) NOT NULL,
                    Predicted_Field VARCHAR(25) NOT NULL,
                    User_level VARCHAR(30) NOT NULL,
                    Actual_skills VARCHAR(300) NOT NULL,
                    Recommended_skills VARCHAR(300) NOT NULL,
                    Recommended_courses VARCHAR(600) NOT NULL
                );
                """
    cursor.execute(table_sql)
    connection.close()

    # Normal User functionality
    if choice == 'Normal User':
        pdf_file = st.file_uploader("Choose your Resume", type=["pdf"])
        if pdf_file is not None:
            save_image_path = './Uploaded_Resumes/' + pdf_file.name
            with open(save_image_path, "wb") as f:
                f.write(pdf_file.getbuffer())
            show_pdf(save_image_path)
                                                    resume_data     =
ResumeParser(save_image_path).get_extracted_data()

            if resume_data:
                st.header("**Resume Analysis**")
                st.success("Hello " + resume_data['name'])
                st.subheader("**Your Basic info**")
                try:
                    st.text('Name: ' + resume_data['name'])
                    st.text('Email: ' + resume_data['email'])
                    st.text('Contact: ' + resume_data['mobile_number'])
                                            st.text('Resume  pages:  '  +
str(resume_data['no_of_pages']))

                            skills_df = pd.DataFrame(resume_data["skills"],
columns=["SKILLS"])
                    st.table(skills_df)

                                                    experience_df   =
pd.DataFrame(resume_data["experience"], columns=["EXPERIENCES"])
                    st.table(experience_df)
```

```
                except KeyError:
                    pass

                cand_level = ''
                if resume_data['no_of_pages'] == 1:
                    cand_level = "Fresher"
                        st.markdown('<h4 style="text-align: left; color:
#d73b5c;">You are looking Fresher.</h4>', unsafe_allow_html=True)
                elif resume_data['no_of_pages'] == 2:
                    cand_level = "Intermediate"
                        st.markdown('<h4 style="text-align: left; color:
#1ed760;">You are at intermediate level!</h4>', unsafe_allow_html=True)
                elif resume_data['no_of_pages'] >= 3:
                    cand_level = "Experienced"
                        st.markdown('<h4 style="text-align: left; color:
#fba171;">You are at experience level!</h4>', unsafe_allow_html=True)

                # Insert data into database
                    insert_data(resume_data['name'], resume_data['email'],
'NA',   str(datetime.datetime.now()),   resume_data['no_of_pages'],   'NA',
cand_level, str(resume_data['skills']), 'NA', 'NA')


                ##  recommendation
                  ds_keyword = ['tensorflow', 'keras', 'pytorch', 'machine
learning', 'deep Learning', 'flask',
                'streamlit']
                  web_keyword = ['react', 'django', 'node jS', 'react js',
'php', 'laravel', 'magento', 'wordpress',
                'javascript', 'angular js', 'c#', 'flask']
                     android_keyword = ['android', 'android development',
'flutter', 'kotlin', 'xml', 'kivy']
                ios_keyword = ['ios', 'ios development', 'swift', 'cocoa',
'cocoa touch', 'xcode']
                     uiux_keyword = ['ux', 'adobe xd', 'figma', 'zeplin',
'balsamiq', 'ui', 'prototyping', 'wireframes',
                                              'storyframes', 'adobe photoshop',
'photoshop', 'editing', 'adobe illustrator',
                                          'illustrator', 'adobe after effects',
'after effects', 'adobe premier pro',
```

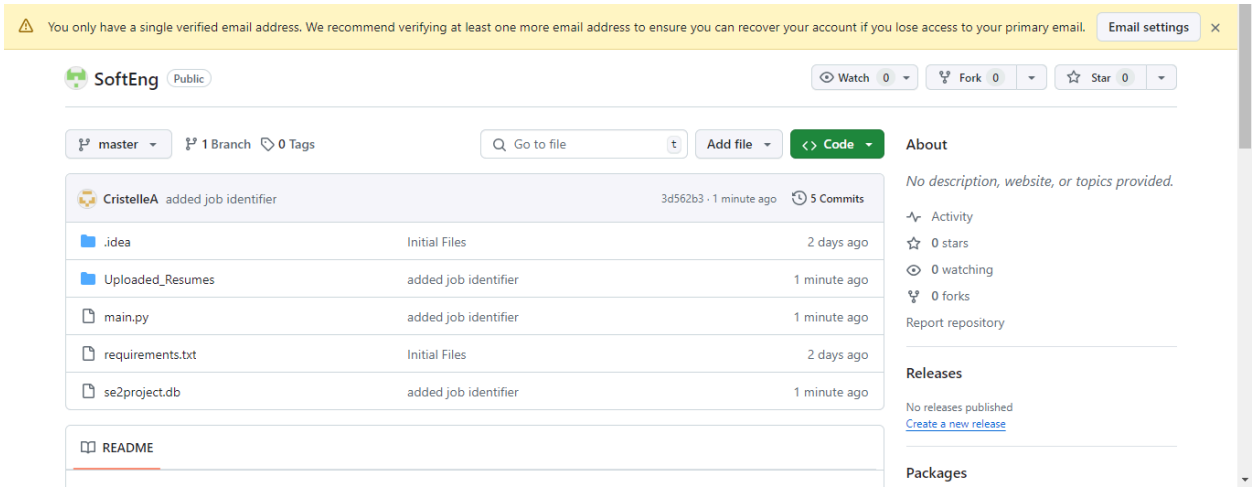
```
                                        'premier pro', 'adobe indesign',
'indesign', 'wireframe', 'solid', 'grasp',
                                    'user research', 'user experience']

                recommended_skills = []
                reco_field = ''
                rec_course = ''
                ## Courses recommendation
                for i in resume_data['skills']:
                    ## Data science recommendation
                    if i.lower() in ds_keyword:
                        print(i.lower())
                        reco_field = 'Data Science'
                          st.success("** Our analysis says you are looking
for Data Science Jobs.**")
                                    recommended_skills = ['Data Visualization',
'Predictive Analysis', 'Statistical Modeling',
                                                'Data Mining', 'Clustering &
Classification', 'Data Analytics',
                                                    'Quantitative Analysis',
'Web Scraping', 'ML Algorithms', 'Keras',
                                                'Pytorch', 'Probability',
'Scikit-learn', 'Tensorflow', "Flask",
                                                'Streamlit']
                        break

                    ## Web development recommendation
                    elif i.lower() in web_keyword:
                        print(i.lower())
                        reco_field = 'Web Development'
                          st.success("** Our analysis says you are looking
for Web Development Jobs **")
                             recommended_skills = ['React', 'Django', 'Node
JS', 'React JS', 'php', 'laravel', 'Magento',
                                                'wordpress', 'Javascript',
'Angular JS', 'c#', 'Flask', 'SDK']

                        st.markdown(
```

```
                                    '''<h4 style='text-align: left; color:
#1ed760;'>Adding this skills to resume will boost🚀 the chances of getting
a Job💼</h4>''',
                                unsafe_allow_html=True)
                            break

                    ## Android App Development
                    elif i.lower() in android_keyword:
                        print(i.lower())
                        reco_field = 'Android Development'
                          st.success("** Our analysis says you are looking
for Android App Development Jobs **")
                                    recommended_skills = ['Android', 'Android
development', 'Flutter', 'Kotlin', 'XML', 'Java',
                                                          'Kivy', 'GIT', 'SDK',
'SQLite']
                        break

                    ## IOS App Development
                    elif i.lower() in ios_keyword:
                        print(i.lower())
                        reco_field = 'IOS Development'
                          st.success("** Our analysis says you are looking
for IOS App Development Jobs **")
                             recommended_skills = ['IOS', 'IOS Development',
'Swift', 'Cocoa', 'Cocoa Touch', 'Xcode',
                                                   'Objective-C', 'SQLite',
'Plist', 'StoreKit', "UI-Kit", 'AV Foundation',
                                                  'Auto-Layout']
                        break

                    ## Ui-UX Recommendation
                    elif i.lower() in uiux_keyword:
                        print(i.lower())
                        reco_field = 'UI-UX Development'
                          st.success("** Our analysis says you are looking
for UI-UX Development Jobs **")
                             recommended_skills = ['UI', 'User Experience',
'Adobe XD', 'Figma', 'Zeplin', 'Balsamiq',
```

```
                                                'Prototyping', 'Wireframes',
'Storyframes', 'Adobe Photoshop', 'Editing',
                                                    'Illustrator', 'After
Effects', 'Premier Pro', 'Indesign', 'Wireframe',
                                                'Solid', 'Grasp', 'User
Research']
                        break


if __name__ == '__main__':
    run()
```

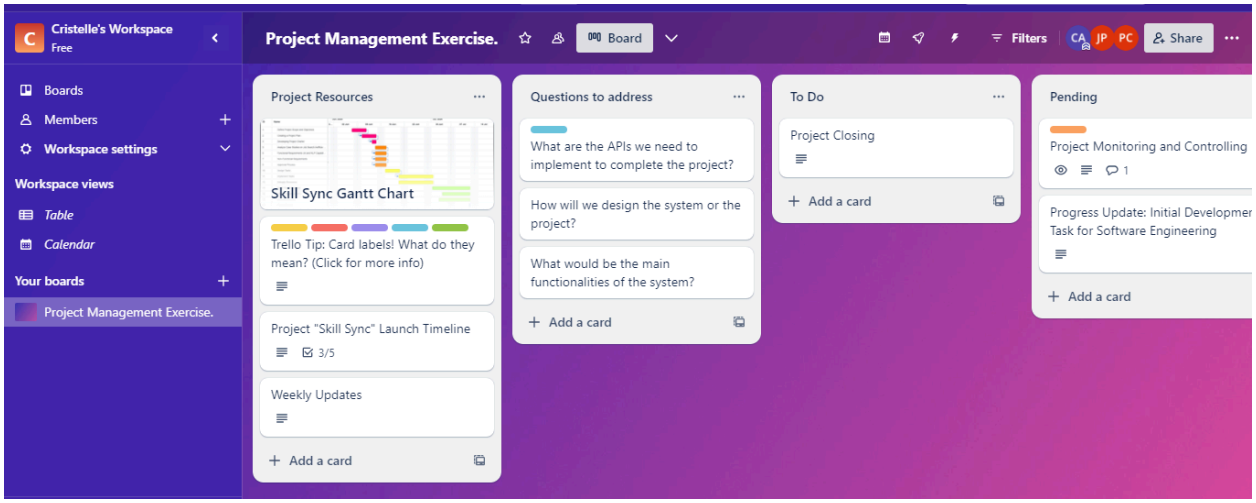
Version Control

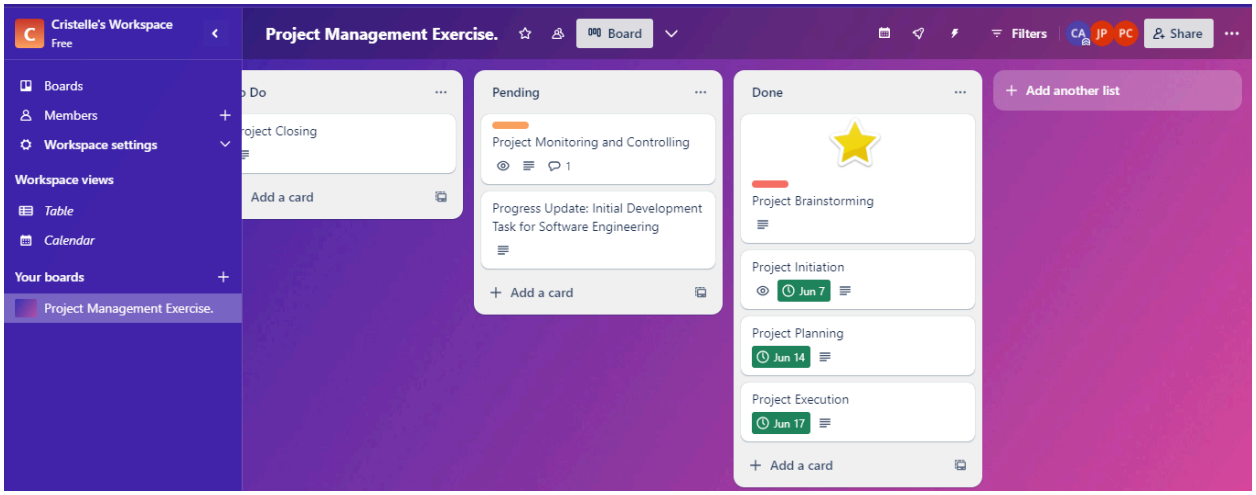
Github Repository: https://github.com/Jayalle/SoftEng.git



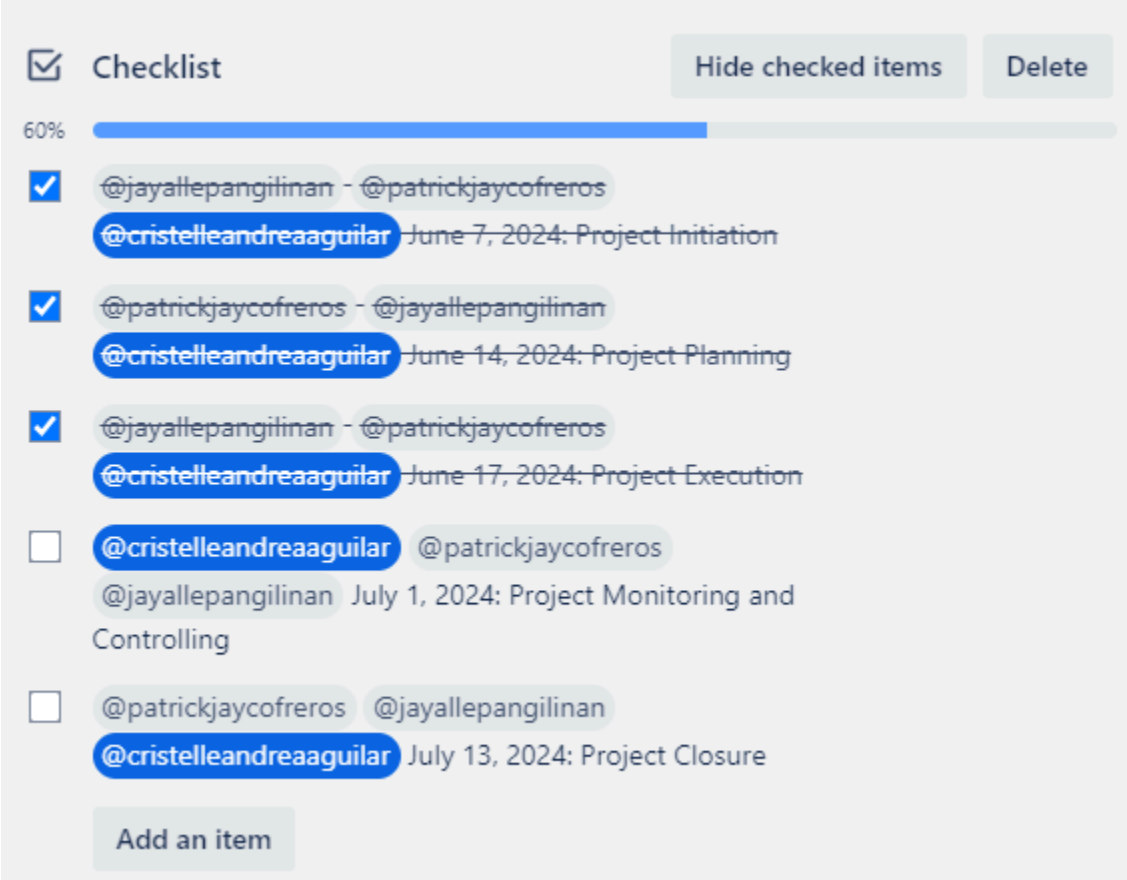
Project Management

Trello link: https://trello.com/invite/b/eyd5XwkU/ATTI5e0caa858f43efc6e95b9c028d5b64dfF8D734D4/project-management-exercise





Updates today:



Week 1

@cristelleandreaaguilar The team were able to complete the system requirements and brainstorm for the possible project ideas.

Week 2

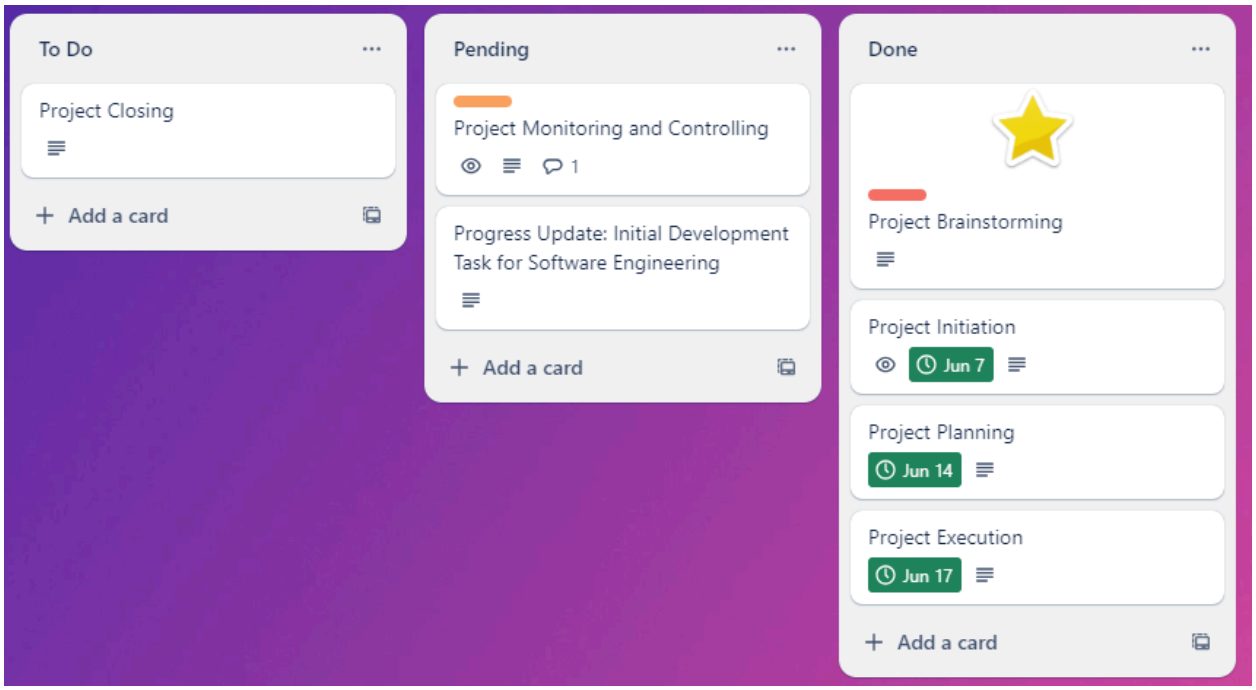
@cristelleandreaaguilar The team were able to complete the project specification, initial documentation and assigning of tasks. The team are also able to gather resources and reference to complete the project.

Week 3

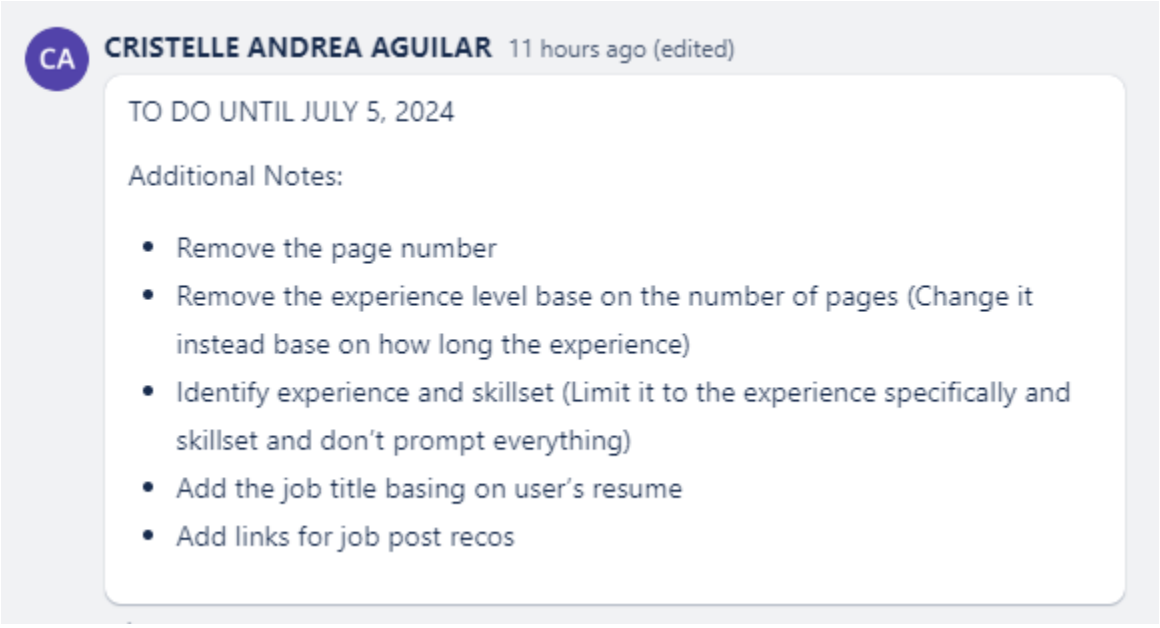
@cristelleandreaaguilar The team were able to collect the dataset and start the initial code.

Features added

- Uploading of resume
- Identifying the skills and experience of the user



Pending tasks:



Transferred project execution to done and move project monitoring and controlling to pending items as well as the task today which is completing the initial documentation.