A Real-time Research Project Report on

AI Resume Analyzer

submitted in partial fulfillment for award of the degree of Bachelor of Technology in

Computer Science and Engineering

by

K Deepika	**22321A6612**
I Meghana	**22321A6628**
N Tareni	**22321A6655**

Under the esteemed

Guidance of

Coordinator

N Sudha Laxmaiah

Assistant Professor , CSE(AIML)

Bhoj Reddy Engineering College for Women

Department of Computer Science and Engineering(AIML)

(Sponsored by Sangam Laxmibai Vidyapeet, approved by AICTE & affiliated o JNTUH)

Vinaynagar, IS Sadan Crossroads, Hyderabad – 500 059

Ph: +91-40-2453 7282; Website: www.brecw.ac.in; Email: principal@brecw.ac.in

2023- 2024

Bhoj Reddy Engineering College for Women

(Sponsored by Sangam Laxmibai Vidyapeet, approved by AICTE & affiliated to JNTUH)
Vinaynagar, I S Sadan Crossroads, Hyderabad 500 059, Telangana
Telephone: 040-2453 7282, Website: www.brecw.ac.in, Email: principal@brecw.ac.in

Date: 05-7-2024

CERTIFICATE

This is to certify that the Real-time Research Project Report entitled **"AI Resume Analyzer"** is a bonafide work carried out by

K Deepika	**22321A6612**
I Meghana	**22321A6628**
N Tareni	**22321A6655**

in partial fulfillment for award of the degree of Bachelor of Technology in Computer Science and Engineering(AIML) department from Bhoj Reddy Engineering College for Women, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH).

Coordinator
N Sudha Laxmaiah

Head of the Department
Dr M Seshu Bhavani

External Examiner

Bhoj Reddy Engineering College for Women

(Sponsored by Sangam Laxmibai Vidyapeet, approved by AICTE & affiliated to JNTUH)
Vinaynagar, I S Sadan Crossroads, Hyderabad 500 059, Telangana
Telephone: 040-2453 7282, Website: www.brecw.ac.in, Email: principal@brecw.ac.in

Date:

CERTIFICATE

This is to certify that the Real-time Research Project Report entitled **"AI RESUME ANALYZER"** is a bonafide work carried out by

K Deepika	**22321A6612**
I Meghana	**22321A6628**
N Tareni	**22321A6655**

in partial fulfillment for award of the degree of Bachelor of Technology in Computer Science and Engineering (AIML)department from Bhoj Reddy Engineering College for Women, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH).

Coordinator	Head of the Department	Principal
N Sudha Laxmaiah	**Dr M Seshu Bhavani**	**Dr J Madhavan**

Sangam Laxmibai Vidyapeet is an educational society for promotion of education among girls and women.It is established in 1952 and registered under the Telangana Societies Registration Act.

ACKNOWLEDGMENT

It is our pleasure to express our whole-hearted thanks to Dr M Seshu Bhavani, Professor & Head of Computer Science and Engineering(AIML) Department for providing the necessary facilities to undergo this project.

We are thankful to our project coordinator N Sudha Laxmaiah, Assistant Professor, Computer Science and Engineering(AIML) Department, for giving us this opportunity to do this project.

We are also thankful to our project reviewer Dr E Madhusudhana, Professor, Computer Science and Engineering(AIML) Department for her extreme guidance and support in completing this project successfully.

We are also thankful to the staff members of Computer Science and Engineering Department(AIML), my friends and to our parents who helped us in completing this project successfully.

By

K Deepika (22321A6612)

I Meghana (22321A6628)

N Tareni (22321A6655)

Contents

List Of figures

Abstract

The aim of this project is to design and develop a tool that results into an easy and helpful solution forapplicants as well as recruiters . “AI RESUME ANALYZER” which parses information from a resume using natural language processing, finds the keywords, cluster them onto sectors based on their keywords And lastly show the recommendation, prediction, analytics to the applicant / recruiter based on keywordmatching.

1. Introduction

1.1 Introduction of the project

Corporate companies and recruitment agencies process numerous resumes daily. This is no task for humans. An automated intelligent system is required which can take out all the vital information from the unstructured resumes and transform all of them to a common structured format which can then be ranked for a specific job position.

Parsed information includes (name, email address, phone number, work experiences, education, hobbies, interests, achievements, certifications, projects) keywords and finally the cluster of theresume (ex: Web Development, Data Science etc.). The parsed information is then stored in a database(MySQL in this case) for later use.

Unlike other unstructured data (ex: email body, web page contents, etc.), resumes are a bit structured. Information is stored in discrete sets. Each set contains data about the person's contact, work experience or education details. In spite of this, resumes are difficult to parse. This is because they vary in types of information, their order, writing style, etc. To parse the data from different kinds of resumes effectively and efficiently, the model must not rely on the order or type of data.

To solve this tedious process our tool comes into action which makes the process fast, easy and reliable. Using NLP Techniques, it extracts keywords from the resume and use it for predictions, recommendation and analytical representation.

The aim of this project is to design and develop a tool that results into an easy and helpful solution for applicants as well as recruiters .“AI RESUME ANALYZER” which parses information from a resume using natural language processing, finds the keywords, cluster them onto sectors based on their keywords. And lastly show the recommendation, prediction, analytics to the applicant / recruiter based on keyword

matching.

1.2 Existing system

The hiring process has evolved significantly over time. Initially, companies advertised job vacancies in newspapers and on television, with applicants sending resumes by post. The hiring team would manually sort resumes and call candidates for interviews, which was time-consuming. As industries grew, so did hiring needs, leading to the rise of hiring consultancies. Applicants had to upload resumes in specific formats to agency websites, which streamlined shortlisting for companies. However, each agency had its own unique format, creating inefficiencies. To address these issues, an intelligent algorithm was developed to parse information from unstructured resumes, sort them into clusters, and rank candidates effectively.

1.3 Problems in Existing System

- This is no task for humans and time consuming.
- It is a challenging task to handle resumes manually.
- Clashes due to their own unique format.
- Requires individual review of each resume from hiring managers.
- The same amount of time and effort is often expended for candidates who are qualified as the ones who are not.

1.4 Proposed System

The proposed system “AI RESUME ANALYZER” is an applicant cum recruiter-based solution which can be widely used by any organization to analyze and get insights of a resume. The model uses natural language processing to understand the resume and then parse the information from it. Once information is parsed it is stored in the database. Quite productive for applicants because it gives predictions, tips and recommendations based on their resume information. System works proper when the uploaded resume is in traditional chronological format. Insightful for admin/recruiter due to its powerful analytics and informative data which is fetched from

user/applicants resume.

1.5 Advantages of Proposed System

- Tracks and analyzes resumes based on job roles.
- Fast, safe, real-time predictions.
- Does the task within a short timespan.

2. Requirements Analysis

2.1 Functional Requirements

User:-

- Register
- Login
- Create Profile
- Manage Profile
- Upload Resume
- Attend interview

Admin:-

- Login
- Manage users and application
- Manage Resume
- Manage skills
- Manage jobs
- Download user’s data into csv file
- Get user feedback and ratings.

AI Agent:-

- Convert Resume
- Parse files and Generate score
- Recommend Job & Skills

2.2 Non-functional Requirements

- Scalability
- Usability
- Security

- Performance

2.3 Computational Resource Requirements

2.3.1 Hardware Resources

- Processor: Intel Core I5.
- RAM : 4GB.
- Hard Disk : 64 GB SSD.

2.3.2 Software Recources

- Data Base: MySQL
- Programming Language:Python
- Frontend: HTML,CSS,Java Script
- Operating System : Windows 7 or above.

2.4 Life cycle model

We use waterfall model for software development process which has various phases in it.

The sequential phases in Waterfall model are –

Requirements Gathering and Analysis: In this phase, the requirements of the AI RESUME ANALYZER are gathered and documented. This includes identifying the functional and non-functional requirements of the system, and creating a detailed requirements document. The requirements are analyzed and studied to understand the needs of the system.

System Design: In this phase, the system is designed based on the analysis done in the previous phase. This includes creating a detailed design document that outlines the system's architecture, components, and interfaces.

Implementation: In this phase, the system is built based on the design created in the previous phase. This includes writing the code, testing, and integrating the different components of the system.

Testing: In this phase, the system is tested to ensure that it meets the requirements and works as expected. This includes functional testing, performance testing, security testing, and usability testing.

Deployment: In this phase, the system is deployed to the production environment. This includes setting up the production environment, configuring the system, and migrating data from the old system (if applicable).

Maintenance: In this phase, the system is maintained and updated to ensure that it continues to meet the changing needs of the users. This includes fixing defects, making updates and enhancements, and providing support to users.

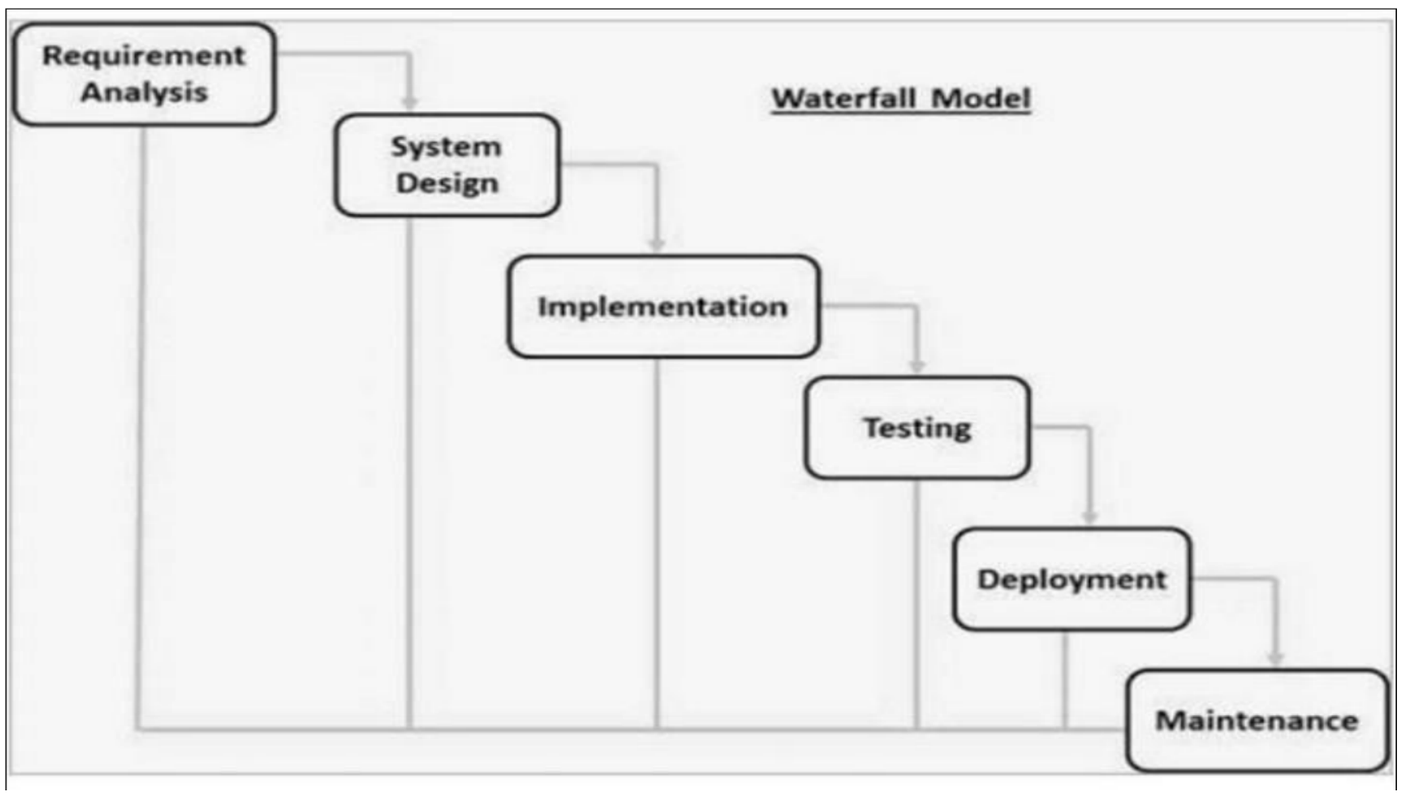


Fig 2.4 Waterfall Model

3. Design

3.1 Architectures

Project architecture represents number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are

(1) Software Architecture

(2) Technical Architecture

3.1.1 Software Architecture

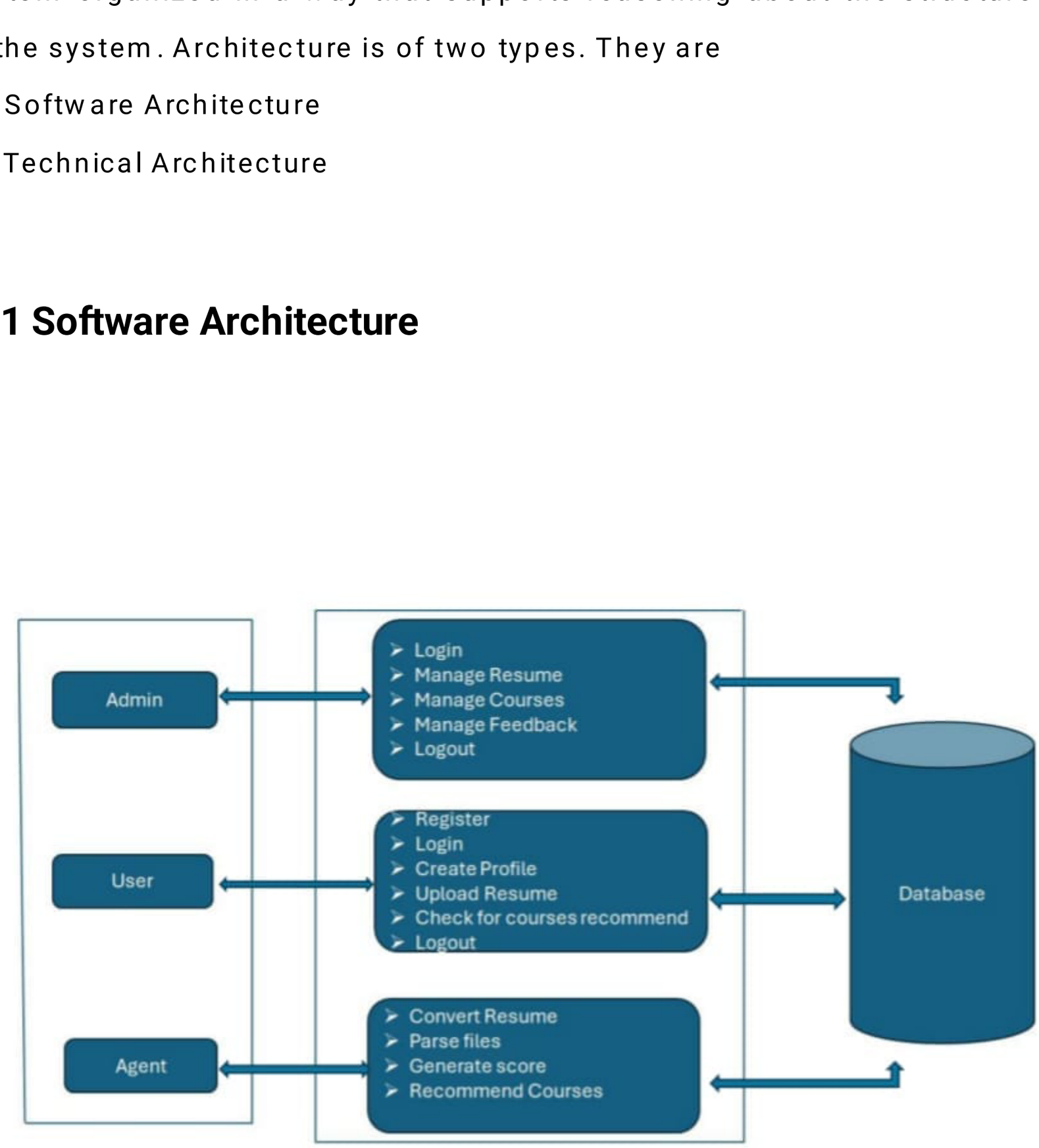


Fig 3.1.1 Software Architecture

3.1.2 Technical Architecture

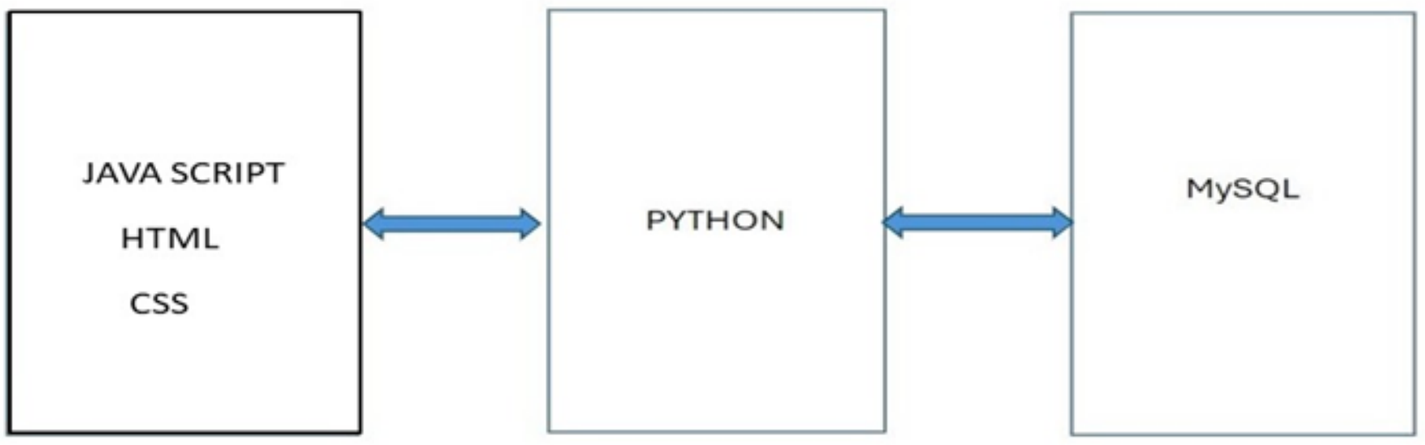


Fig 3.1.2 Technical Architecture

3.2 Usecase Diagram

Use case diagram are used to gather the requirements of a system including internal and external influences. A use case represents a functionality of a system. So, use case diagrams are used to describes the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

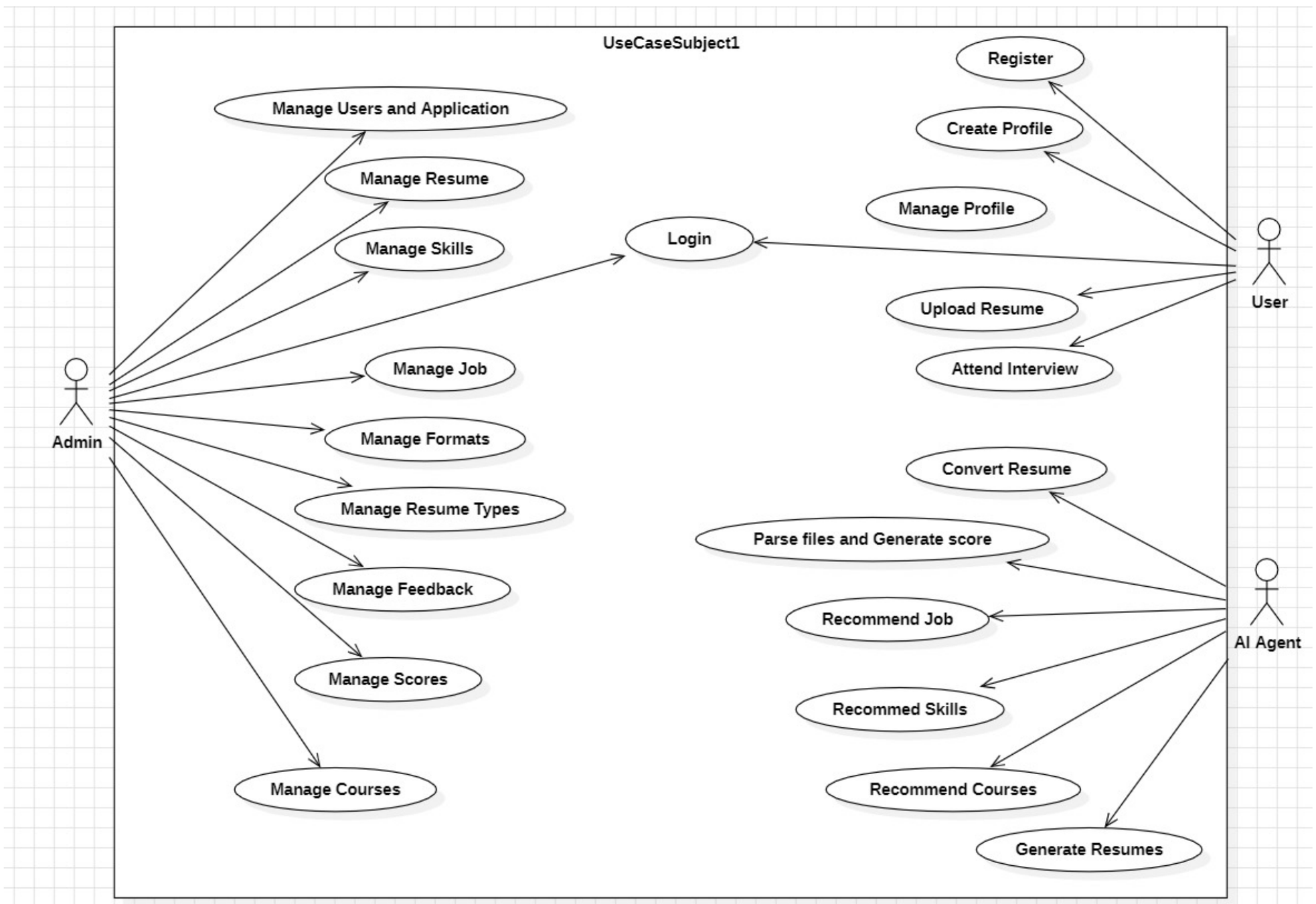


Fig 3.2.1

3.3 Class Diagram

A class diagram in software engineering is a type of static structure diagram that visualizes the structure of a system by showing its classes, their attributes, methods, and the relationships among objects. It includes classes depicted as rectangles divided into three compartments (name, attributes, methods), and illustrates various relationships like associations, inheritances (generalizations), aggregations, compositions, and dependencies. These relationships are represented by different types of connecting lines and symbols, such as solid lines for associations, hollow arrows for inheritances, and diamonds for aggregations and compositions. Class diagrams are integral to the Unified Modeling Language (UML) and are essential for understanding, designing, and documenting the blueprint of a software system.

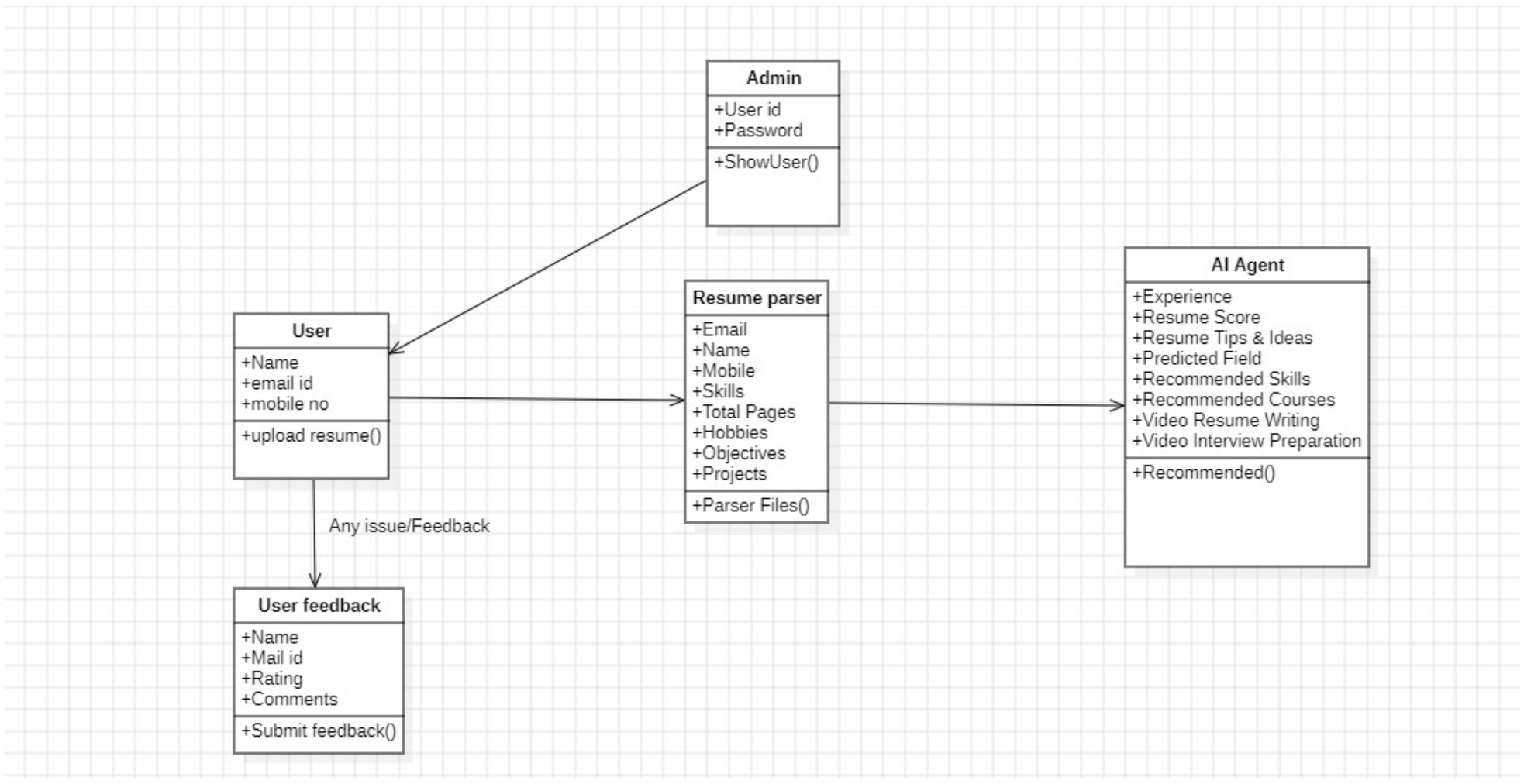


Fig 3.3.1

3.4 Sequence Diagram

A sequence is Unified Modelling Language is a kind of interaction diagram that show how processes operate with another and in what order. It is a construct of a message chart. It shows objects (or classes) as lifelines, depicted as vertical dashed lines, and their interactions as horizontal arrows representing messages exchanged over time. The sequence diagram highlights the order of operations, with time progressing from top to bottom, and includes elements such as activation bars to indicate the duration

of an object's involvement in the interaction. It is particularly useful for modeling the dynamic behavior of a system, specifying detailed workflows, and documenting use case scenarios in the Unified Modeling Language .

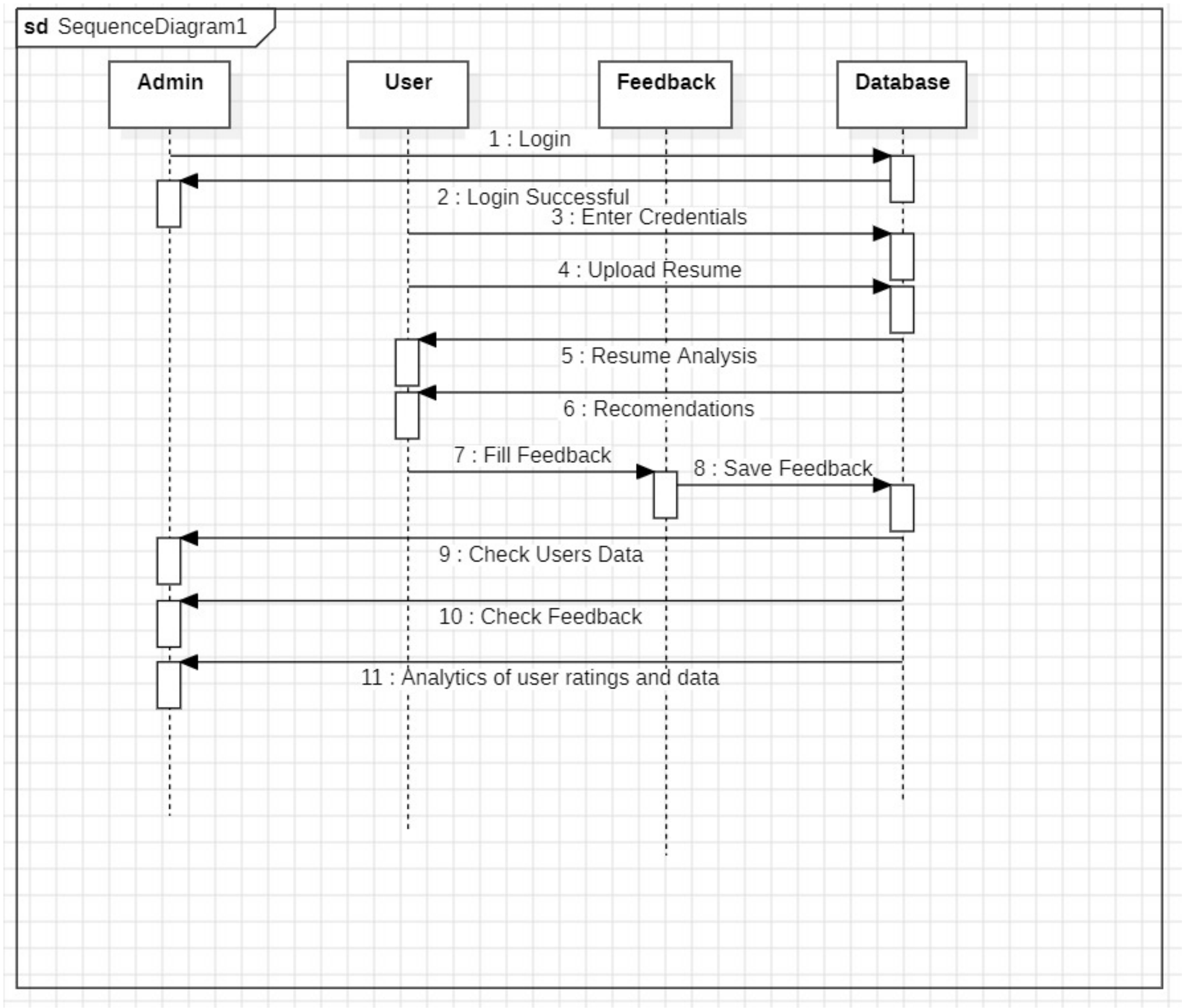


Fig 3.4.1

3.5 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, in software engineering is a type of interaction diagram that emphasizes the structural organization of objects that send and receive messages, illustrating the interactions between objects or parts in a system. Unlike sequence diagrams that focus on the chronological order of messages, collaboration diagrams highlight the relationships and roles of objects in the context of their interactions. Objects are represented as rectangles, and their links are shown as lines connecting them, with messages labeled on these

lines to indicate the communication between them. Collaboration diagrams are valuable for visualizing the flow of control and data among components, making them useful for detailing and documenting complex interactions within the Unified Modeling Language (UML).

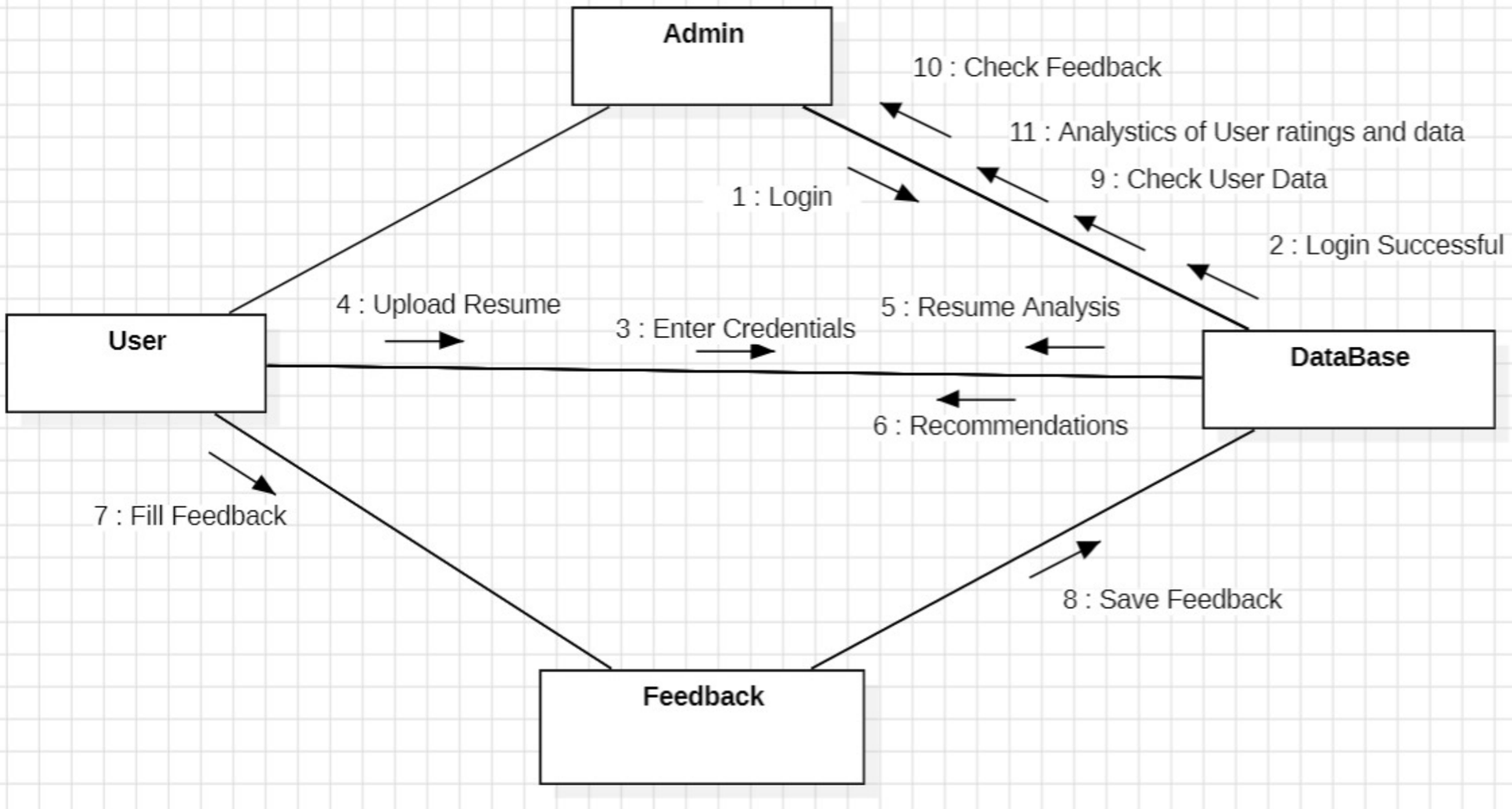


Fig 3.5.1

3.6 Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that visually represents the flow of activities within a system, showcasing the sequence from one activity to another.

Activity diagram for User

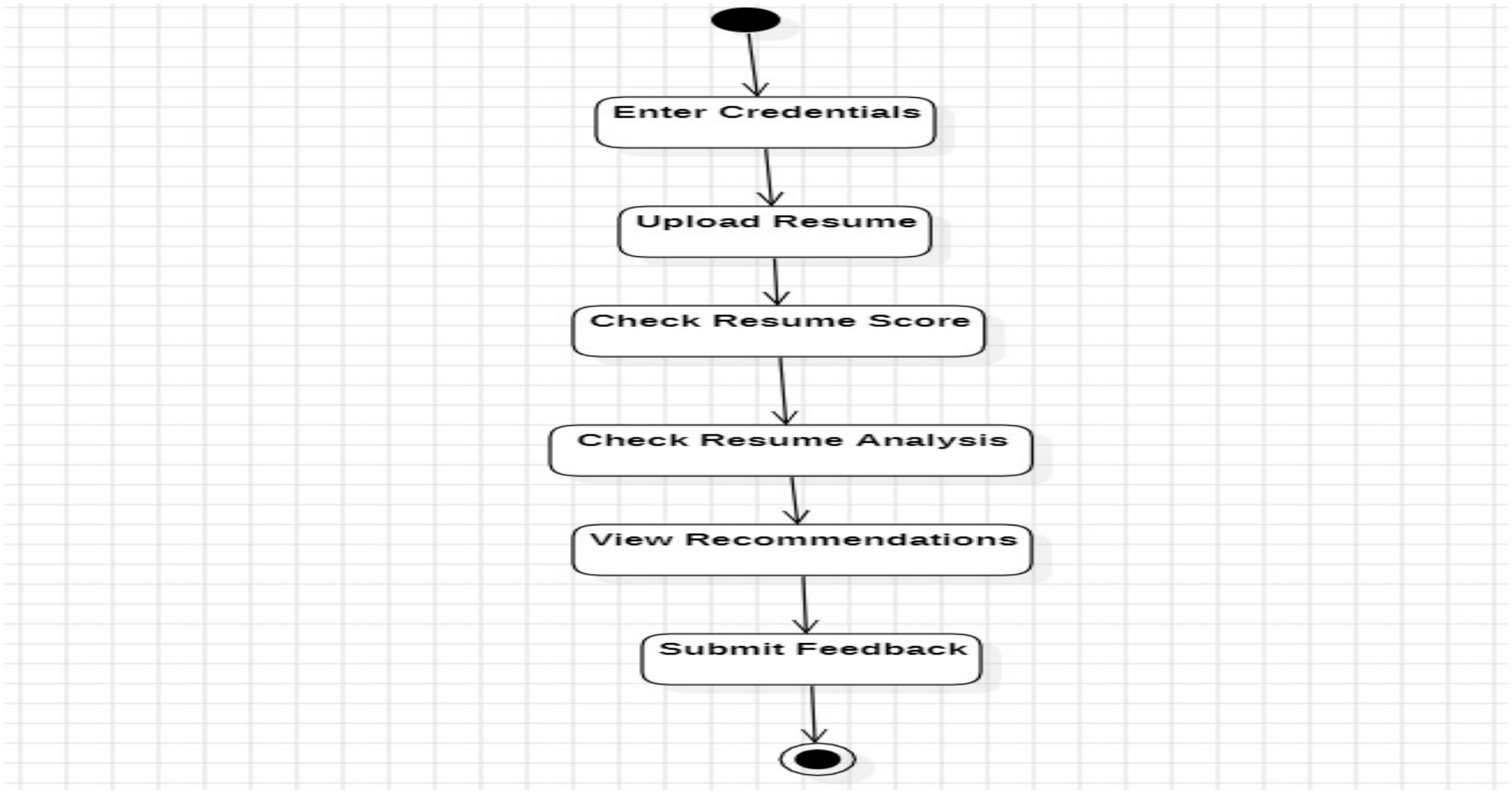


Fig 3.6.1

Activity diagram for Admin

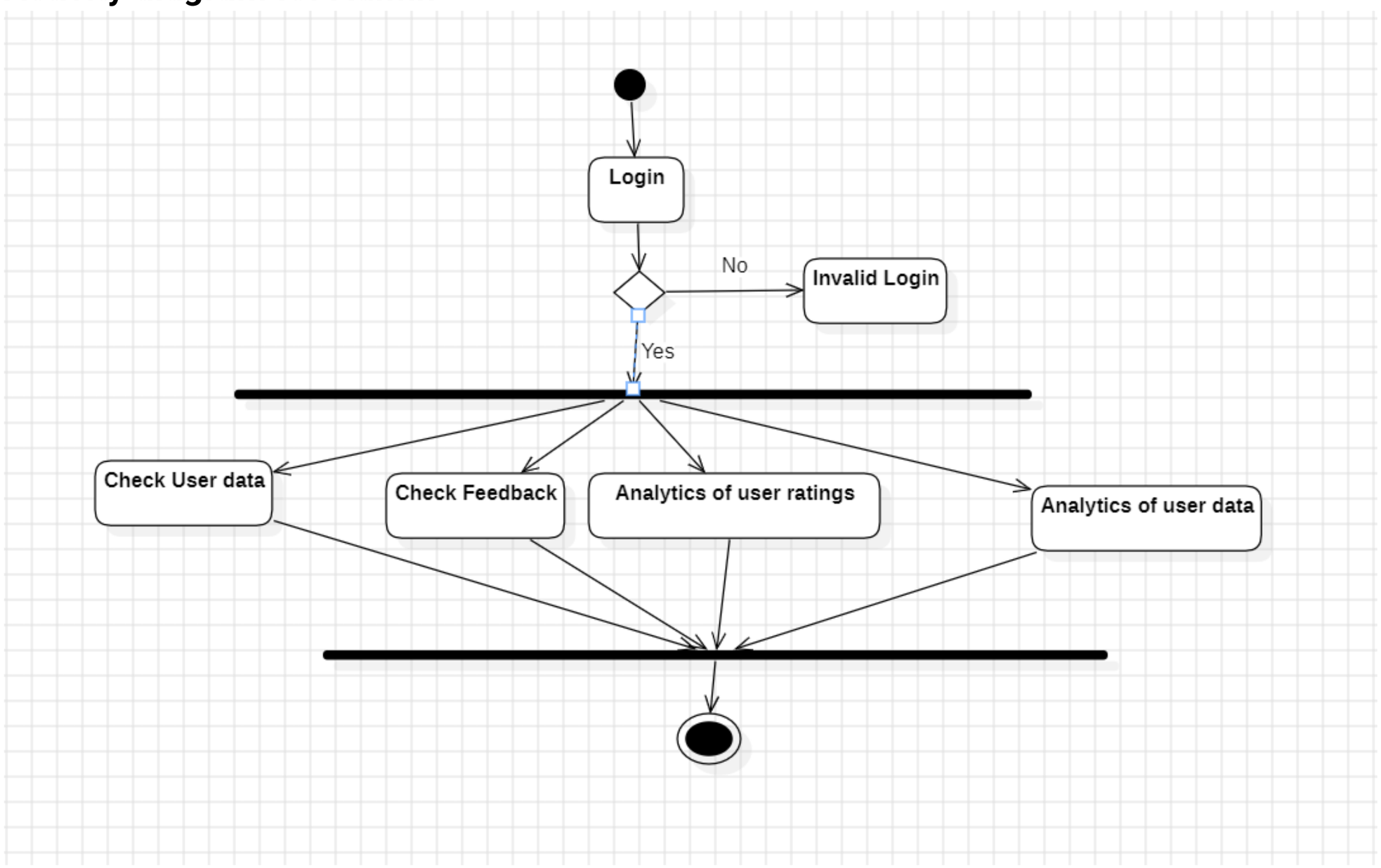


Fig 3.6.2

Activity diagram for AI Agent

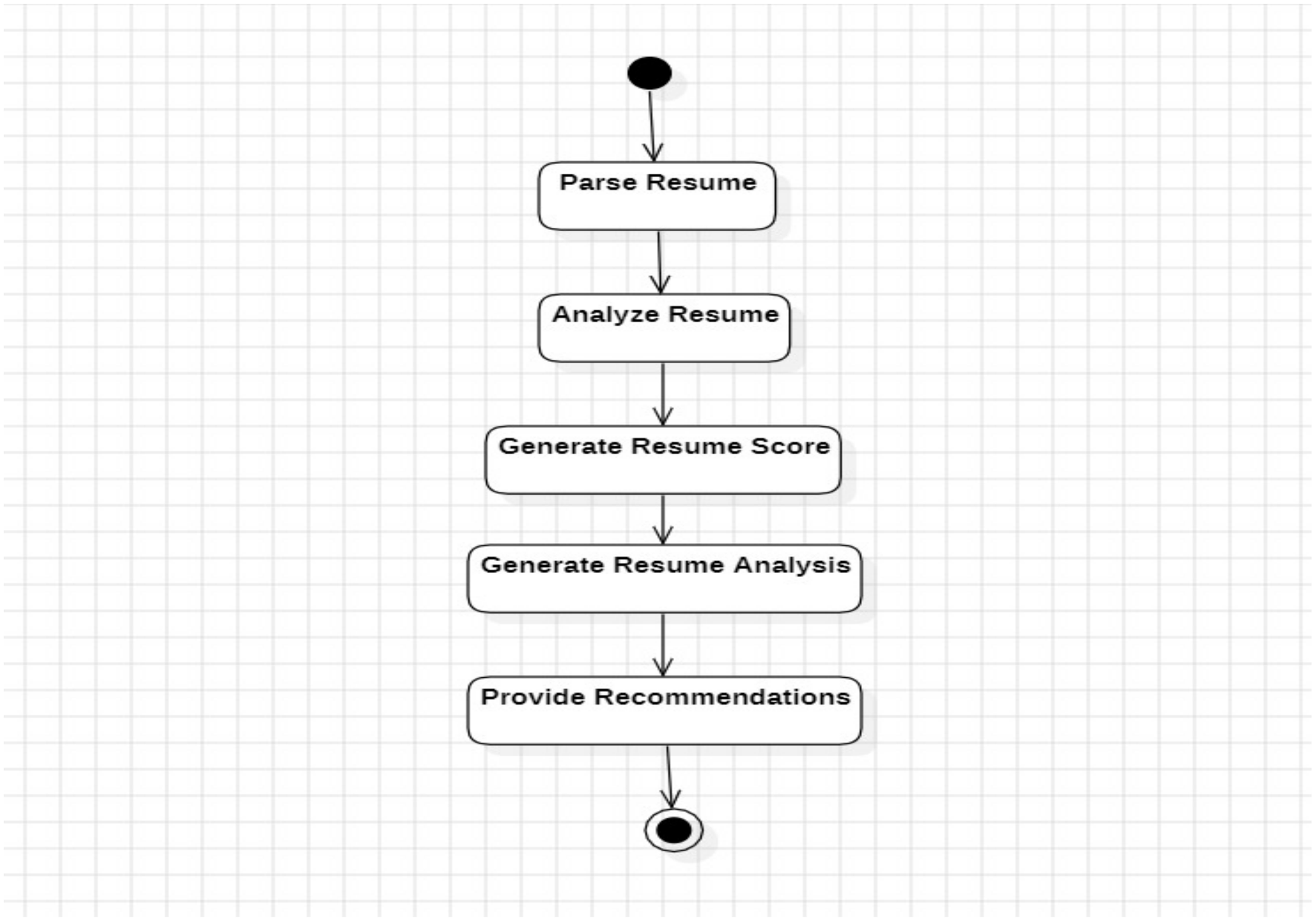


Fig 3.6.3

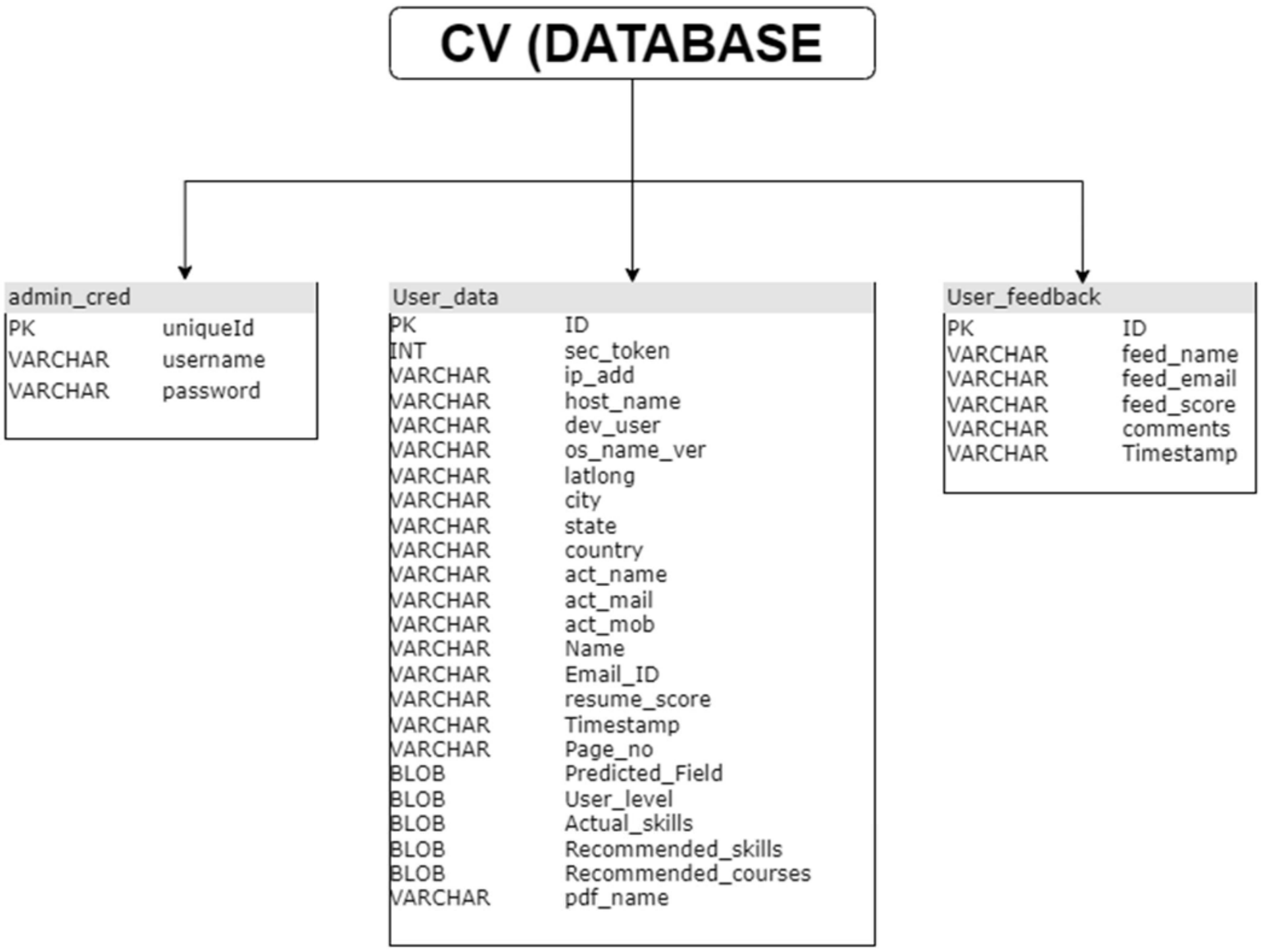
3.7 Database Design

A pictorial representation of how the database and its table is

admin_cred :- Stores data of user credentials

user_data :- Stores data of parsed and fetched information from user and it’s resume

user_feedback :- Stores feedback data provided by user’s



Database Design

4.Implementation

4.1 Technologies:

HTML5 :- HTML is the standard markup language for Web pages. With HTML you can create your own website.

CSS3 :- CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

JavaScript :- JavaScript is the world's most popular programming language. JavaScript is the programming language of the Web.

Streamlit :- Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps.

Python :- Python is a popular programming language. Python can be used on a server to create web applications.

JSON :- JSON is a text format for storing and transporting data. JSON is "self-describing" and easy to understand

MySQL :- MySQL is free and open-source. And is a widely used relational database management system (RDBMS)

4.2 Psuedo Code

```
# Generates a link allowing the data in a given panda dataframe to be downloaded in
csv format def get_csv_download_link(df,filename,text):

csv = df.to_csv(index=False)

## bytes conversions

b64 = base64.b64encode(csv.encode()).decode()

 href = f'<a href="data:file/csv;base64,{b64}" download="{filename}">{text}</a>'

   return href
```

```
# Reads Pdf file and check_extractable

def pdf_reader(file):

    resource_manager = PDFResourceManager()

    fake_file_handle = io.StringIO()

                converter = TextConverter(resource_manager, fake_file_handle,
laparams=LAParams())

    page_interpreter = PDFPageInterpreter(resource_manager, converter)

    with open(file, 'rb') as fh:

        for page in PDFPage.get_pages(fh,

                                caching=True,

                                check_extractable=True):

            page_interpreter.process_page(page)

            print(page)

        text = fake_file_handle.getvalue()

    ## close open handles

    converter.close()

    fake_file_handle.close()

    return text

# course recommendations which has data already loaded from Courses.py

def course_recommender(course_list):

    st.subheader("**Courses & Certificates Recommendations 👨🎓**")

    c = 0

    rec_course = []

    ## slider to choose from range 1-10
```

```
no_of_reco = st.slider('Choose Number of Course Recommendations:', 1, 10, 5)
random.shuffle(course_list)
for c_name, c_link in course_list:
    c += 1
    st.markdown(f"({c}) [{c_name}]({c_link})")
    rec_course.append(c_name)
    if c == no_of_reco:
        break
  return rec_course
###### Database ######
# sql connector
connection =
pymysql.connect(host='localhost',user='root',password='jZ#9rhsw',db='cv')
cursor = connection.cursor()
# inserting miscellaneous data, fetched results, prediction and recommendation into
user_data table
def
insert_data(sec_token,ip_add,host_name,dev_user,os_name_ver,latlong,city,state,count
ry,act_name,act_mail,act_mob,name,email,res_score,timestamp,no_of_pages,reco_fiel
d,cand_level,skills,recommended_skills,courses,pdf_name):
  DB_table_name = 'user_data'
  insert_sql = "insert into " + DB_table_name + """
                                                values
(0,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"""
                                                rec_values =
(str(sec_token),str(ip_add),host_name,dev_user,os_name_ver,str(latlong),city,state,cou
```

```
ntry,act_name,act_mail,act_mob,name,email,str(res_score),timestamp,str(no_of_pages
),reco_field,cand_level,skills,recommended_skills,courses,pdf_name)
    cursor.execute(insert_sql, rec_values)
    connection.commit()
cand_level = ''
            if resume_data['no_of_pages'] < 1:
                cand_level = "NA"
                    st.markdown( '''<h4 style='text-align: left; color: #d73b5c;'>You are at
Fresher level!</h4>''',unsafe_allow_html=True)
    #### if internship then intermediate level
            elif 'INTERNSHIP' in resume_text:
                cand_level = "Intermediate"
                    st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at
intermediate level!</h4>''',unsafe_allow_html=True)
            elif 'INTERNSHIPS' in resume_text:
                cand_level = "Intermediate"
                    st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at
intermediate level!</h4>''',unsafe_allow_html=True)
            elif 'Internship' in resume_text:
                cand_level = "Intermediate"
                    st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at
intermediate level!</h4>''',unsafe_allow_html=True)
            elif 'Internships' in resume_text:
                cand_level = "Intermediate"
                    st.markdown('''<h4 style='text-align: left; color: #1ed760;'>You are at
intermediate level!</h4>''',unsafe_allow_html=True)
```

```
#### if Work Experience/Experience then Experience level

            elif 'EXPERIENCE' in resume_text:

               cand_level = "Experienced"

## Skills Analyzing and Recommendation

            st.subheader("**Skills Recommendation 💡**")

### Current Analyzed Skills

            keywords = st_tags(label='### Your Current Skills',

              text='See our skills recommendation below',value=resume_data['skills'],key =
'1 ### Keywords for Recommendations

                        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#', 'Asp.net', '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']

                        n_any = ['english','communication','writing', 'microsoft office',
'leadership','customer management', 'social media']

            ### Skill Recommendations Starts

            recommended_skills = []

            reco_field = ''

## Resume Scorer & Resume Writing Tips
```

```
st.subheader("**Resume Tips & Ideas 🥂**")

resume_score = 0

### Predicting Whether these key points are added to the resume

if 'Objective' or 'Summary' in resume_text:

    resume_score = resume_score+6

        st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added Objective/Summary</h4>''',unsafe_allow_html=True)

        else:

            st.markdown('''<h5 style='text-align: left; color: #000000;'>[-] Please add your career objective, it will give your career intension to the Recruiters.</h4>''',unsafe_allow_html=True)


        if 'Education' or 'School' or 'College'  in resume_text:

            resume_score = resume_score + 12

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added Education Details</h4>''',unsafe_allow_html=True)

        else:

            st.markdown('''<h5 style='text-align: left; color: #000000;'>[-] Please add Education. It will give Your Qualification level to the recruiter</h4>''',unsafe_allow_html=True

 if 'EXPERIENCE' in resume_text:

            resume_score = resume_score + 16

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added Experience</h4>''',unsafe_allow_html=True)

        elif 'Experience' in resume_text:

            resume_score = resume_score + 16
```

```
                st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You
have added Experience</h4>''',unsafe_allow_html=True)

            else:

 st.markdown('''<h5 style='text-align: left; color: #000000;'>[-] Please add Experience. It
will help you to stand out from crowd</h4>''',unsafe_allow_html=True)

        if 'INTERNSHIPS' in resume_text:

            resume_score = resume_score + 6

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You
have added Internships</h4>''',unsafe_allow_html=True)

        elif 'INTERNSHIP' in resume_text:

            resume_score = resume_score + 6

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You
have added Internships</h4>''',unsafe_allow_html=True)

        elif 'Internships' in resume_text:

            resume_score = resume_score + 6

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You
have added Internships</h4>''',unsafe_allow_html=True)

        elif 'Internship' in resume_text:

            resume_score = resume_score + 6

            st.markdown('''<h5 style='text-align: left; color: #1ed760;'>[+] Awesome! You
have added Internships</h4>''',unsafe_allow_html=True)

        else:

            st.markdown('''<h5 style='text-align: left; color: #000000;'>[-] Please add
Internships. It will help you to stand out from crowd</h4>''',unsafe_allow_html=True

# Pie chart for user ratings

        st.subheader("*User Rating's*")
```

```
        fig = px.pie(values=values, names=labels, title="Chart of User Rating Score From
1 - 5 🤗", color_discrete_sequence=px.colors.sequential.Aggrnyl)

        st.plotly_chart(fig)
```

5.Testing

5.1 Overview

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect free in order to produce the quality product.

As per the current trend, due to constant change and development in digitization, our lives are improving in all areas. The way we work is also changed. We access our bank online, we do shop online; we order food online and many more. We rely on software's and systems. What if these systems turnout to be defective? We all know that one small bug shows huge impact on business in terms of financial loss and goodwill. To deliver a quality product, we need to have Software Testing in the Software Development Process.

Some of the reasons why software testing becomes very significant and integral part in the field of information technology are as follows:

1. Cost effectiveness

2. Customer Satisfaction

3. Security

4. Product Quality

5.2 Dimensions of Testing

There are many different dimensions to consider:

1. Layers of the application (database, APIs, UI)

2. Scale of testing (unit, module, integration, scenario)

3. Type of testing (functional, performance, security, etc.)

4. Methodology (exploratory, scripted manual, automated)

5.3 Stages of Testing

5.3.1 Unit Testing

During This first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim of this endeavour is to determine whether the application functions as designed.

In this phase, a unit can refer to a function, individual program or even a procedure, and White box testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It quite common for software developers to perform unit tests before delivering software to testers for formal testing.

5.3.2 Integration Testing

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined.

5.3.3 System Testing

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production.

System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

5.3.4 Acceptance Testing

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. The extensiveness of these tests is just another reason why bringing software testers in early is important. When a program is more thoroughly tested, a greater number of bugs will be detected; this ultimately results in higher quality software.

5.4 Types of testing

5.4.1 Black box testing

It is also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

5.4.2 White box testing

It is also called as Glass Box, Clear Box, Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually done at the unit level.

White Box Testing Techniques:

1.Statement Coverage

2.Branch Coverage

TEST CASE	TEST CASE DESCRIPTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1.Login	Check response when valid email and password is entered	Username: admin Password: admin@-resume-analyzer	Login should be successful	Login was unsuccessful	Fail
2.	Check response when valid email and password is entered	Username: admin Password: admin@resume-analyzer	Login should be successful	Login was successful	Pass
3.	Checks user data and feedback	User data and Feedback	Data Processed	Data Processed	Pass

3.Path Coverage

5.5 Test Cases

Test Case for Admin:

Test Case for User:

TEST CASE	TEST CASE DESCRIPTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1.	Insert Credentials	Name : Nalli Tareni Email Id: Vijayatareni2016@gmail.com Phone no:8074341097	The inserted data should be save in database	Inserted data users saved in successful	Pass
2.	Upload Resume	Resume Uploaded in pdf format with 200MB	The resume should be uploaded	The resume is uploaded successful	Pass
3.	Upload Resume	Resume Uploaded in word format	The resume should be uploaded	The resume is not Allowed	Fail
4.	Check for Resume Score	Resume score generated	Resume score generated	Resume score generated	Pass
5.	Check for	Submit button after editing	Data in profile	Profile was edited	Pass

	Resume	profile	should be editable	successfully	

Test Case for Feedback:

TEST CASE	TEST CASE DESCRIPTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1.	Display past feedback	Past feedback data	Data to be dispalyed	Data displayed	Pass
2.	Enter Feedback	Entered Feedback	Feedback should be saved	Feedback saved successfully	Pass
3.	Check Feedback saved	Feedback status	Feedback should be presented	Feedback presented	Pass

6.Screenshots

1.Homepage

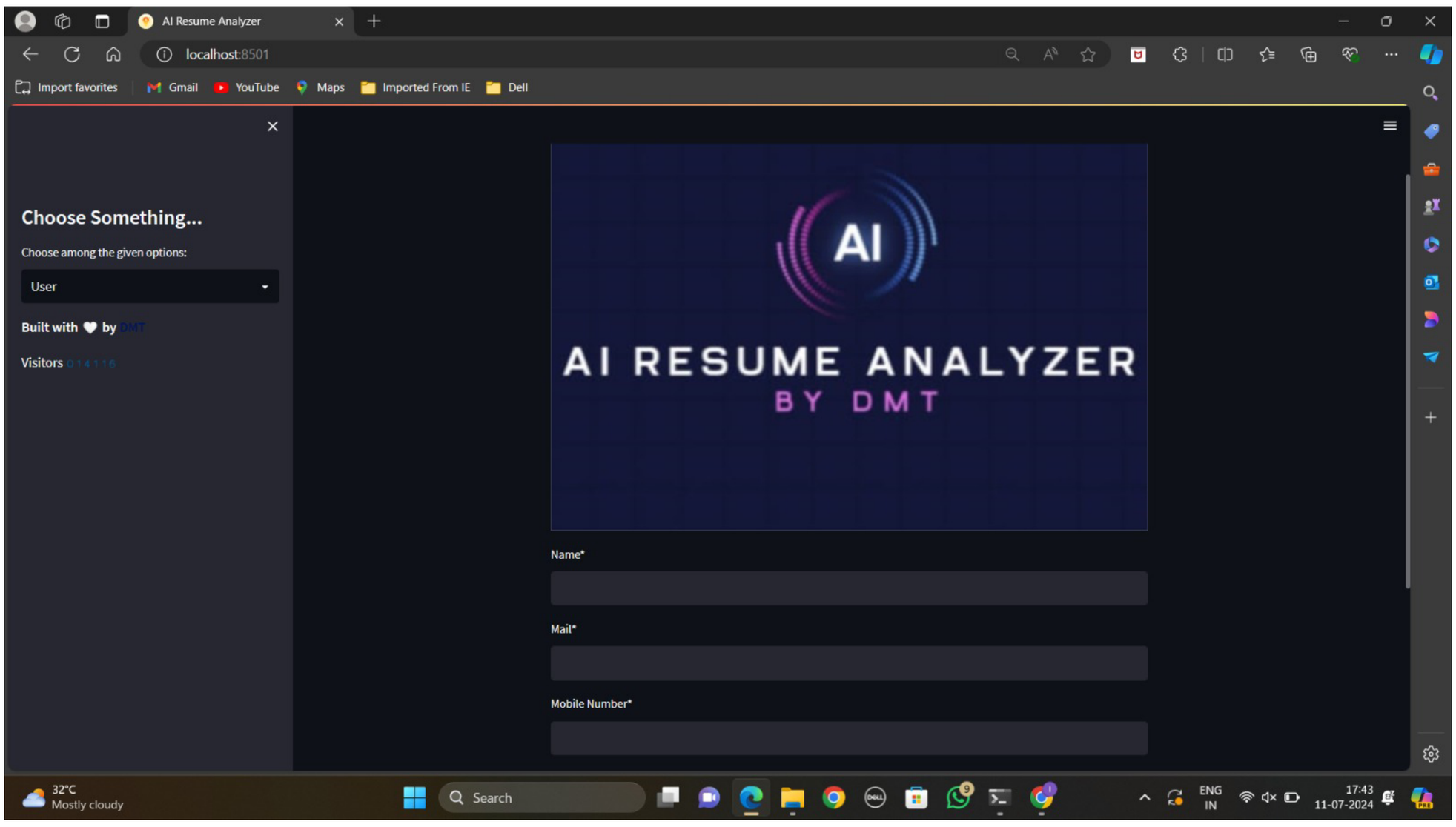


Fig 6.1

2.Upload and view resume

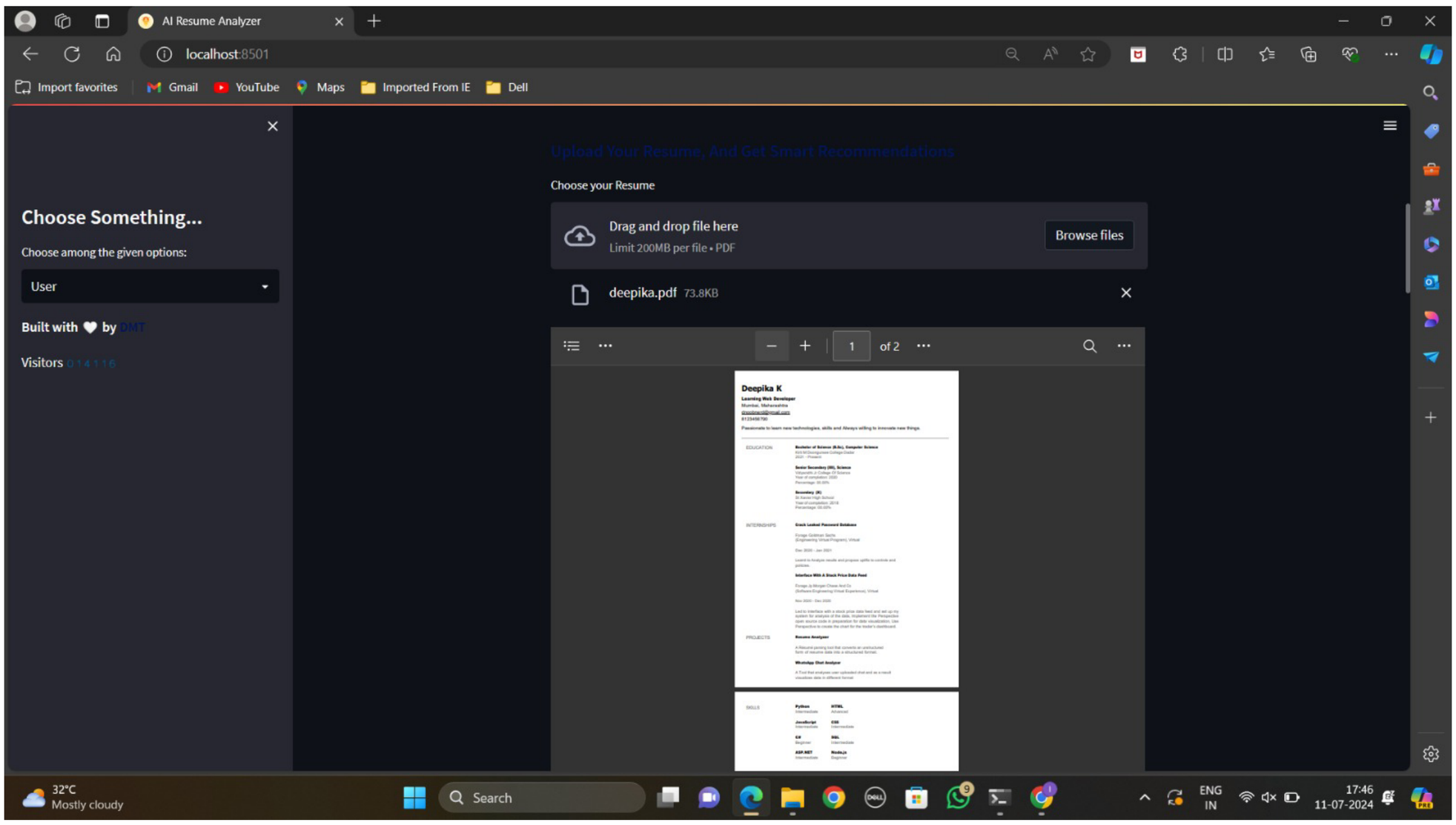


Fig 6.2

3.Analysis

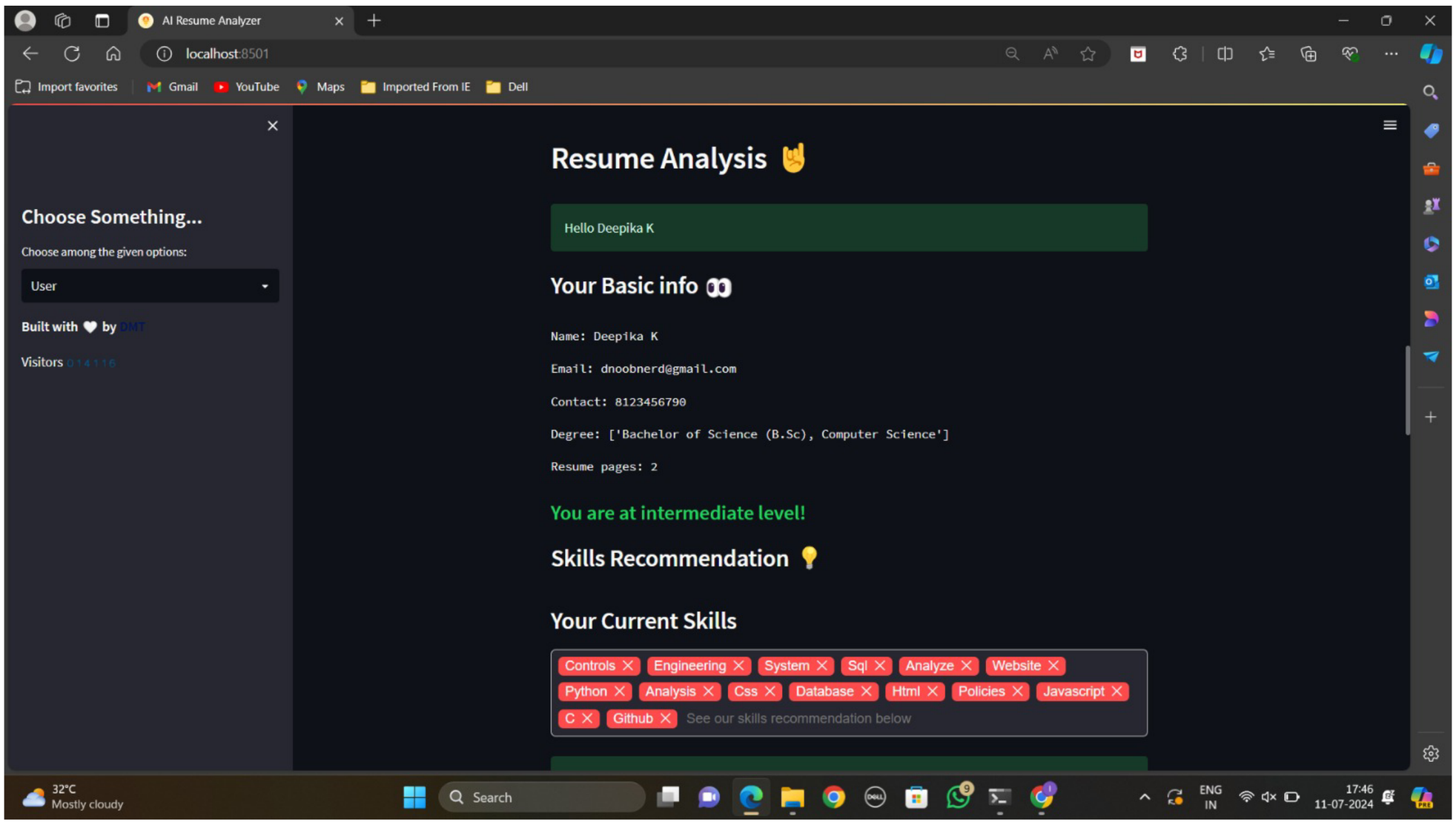


Fig 6.3

4.Recommendation and Prediction

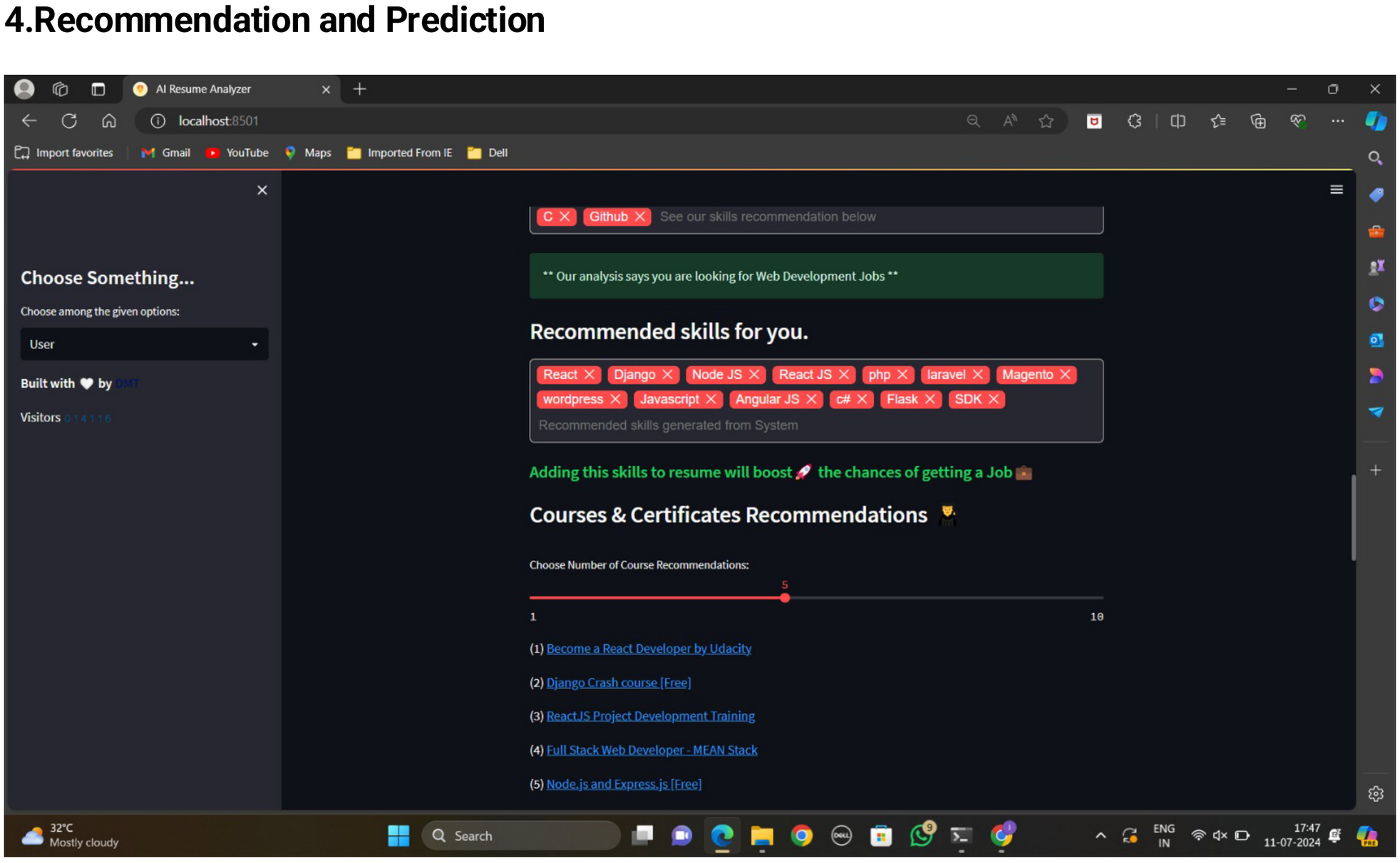


Fig 6.4

5.Resume tips & ideas with overall Scorer

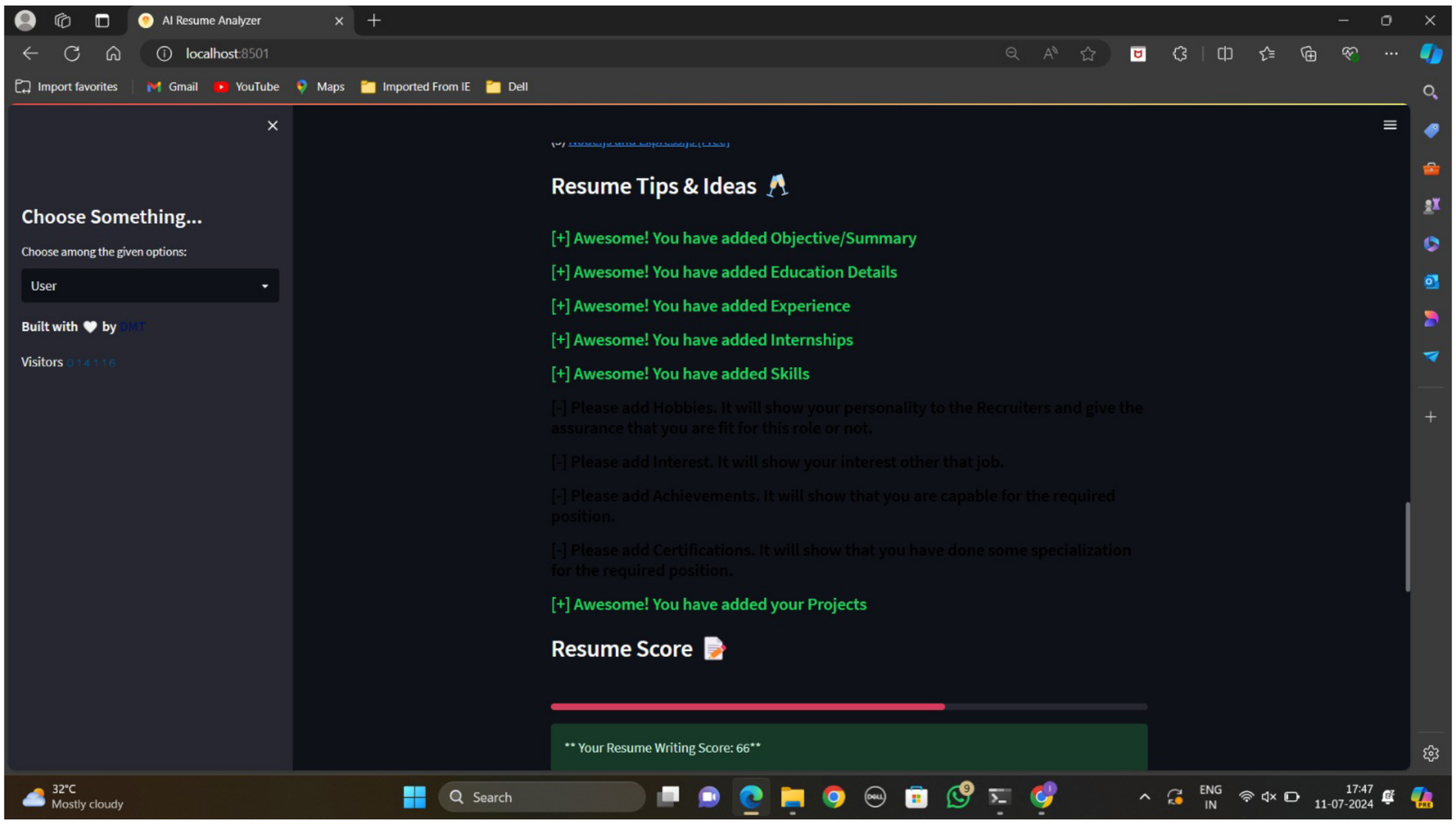


Fig 6.5

6.Feedback Form

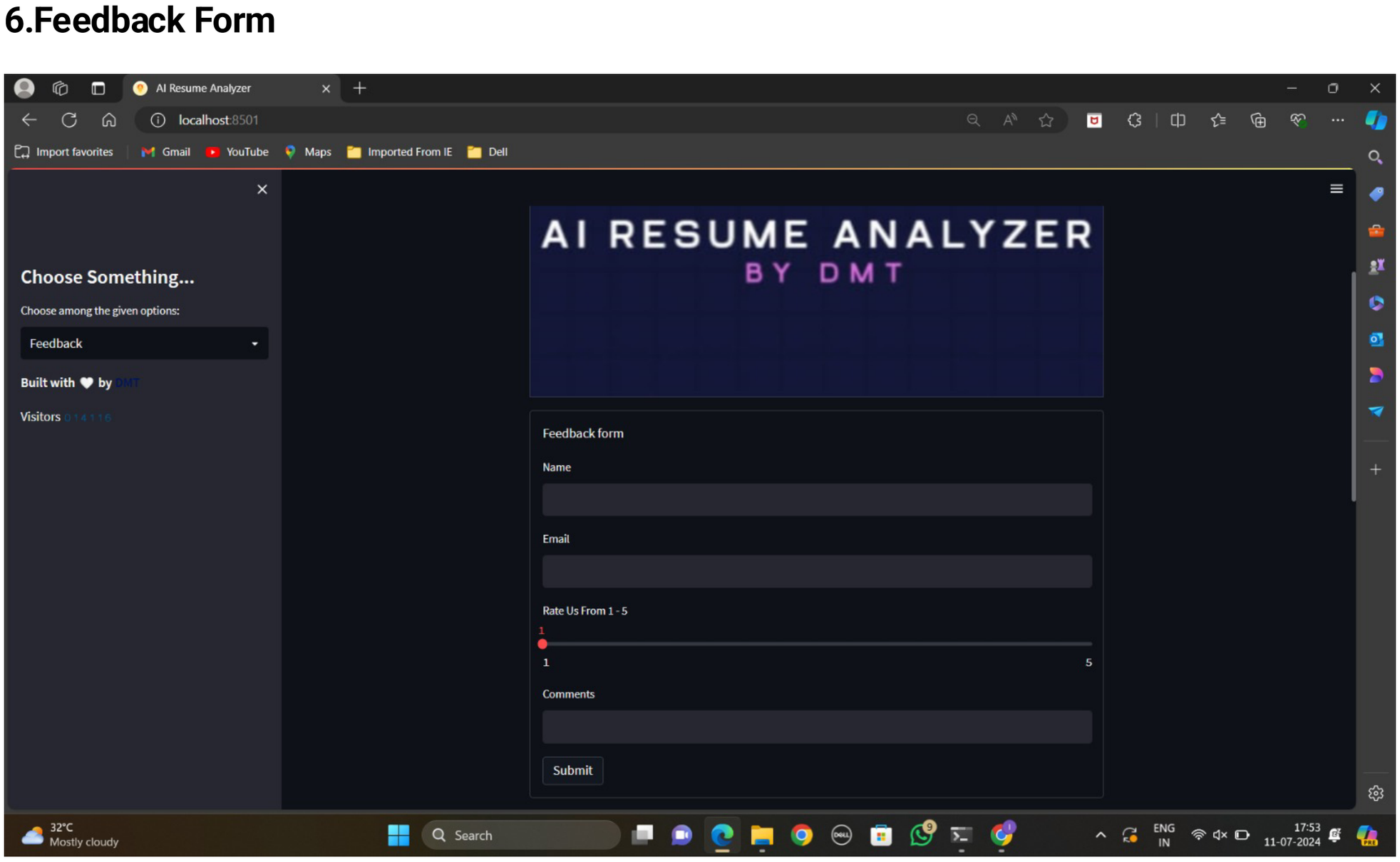


Fig 6.6

7.Past user ratings and comments

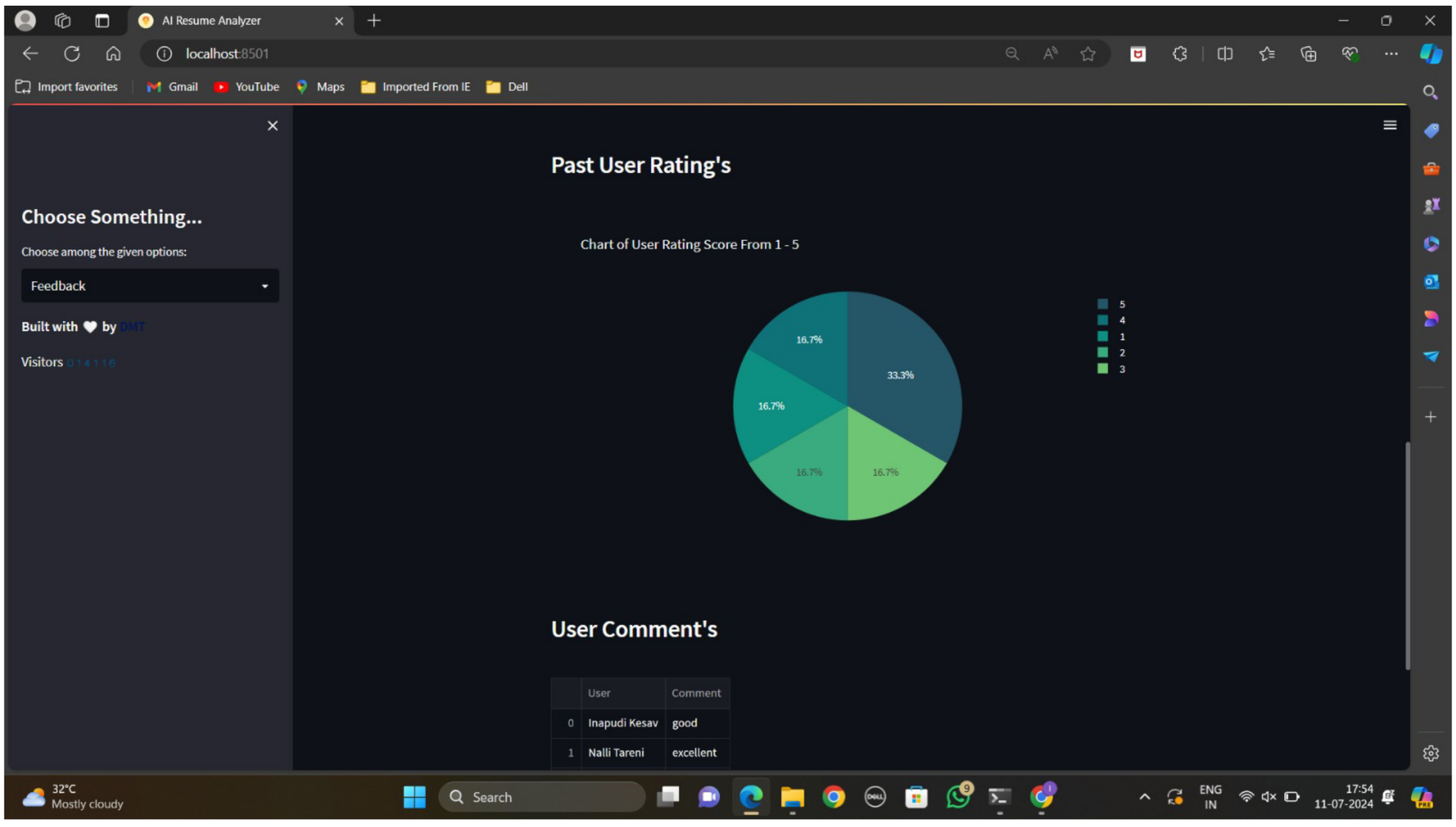


Fig 6.7

8.Admin Login

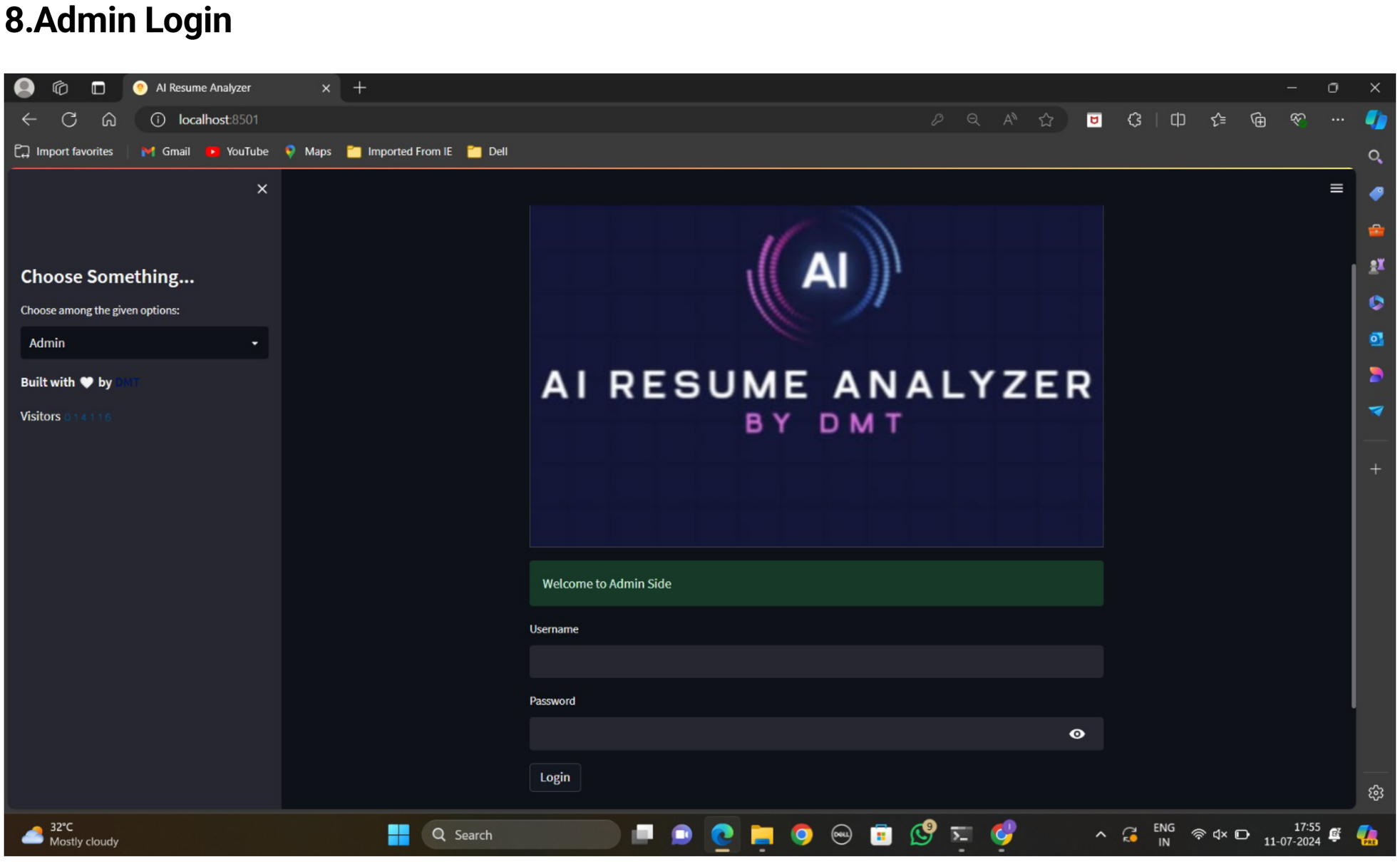


Fig 6.8

9.Total user's user data table, csv file download link, feedback data table

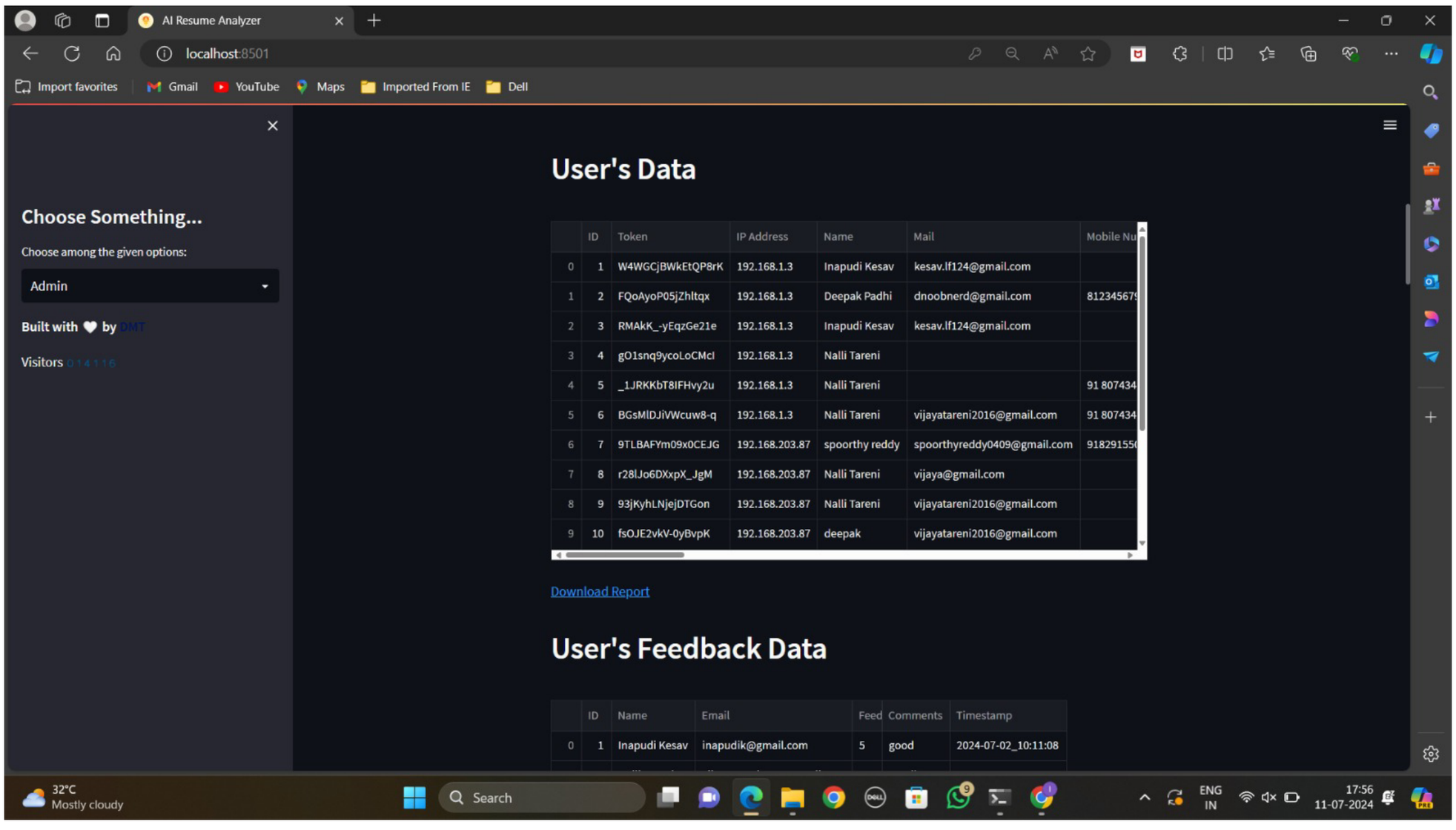


Fig 6.9

10.Download csv file

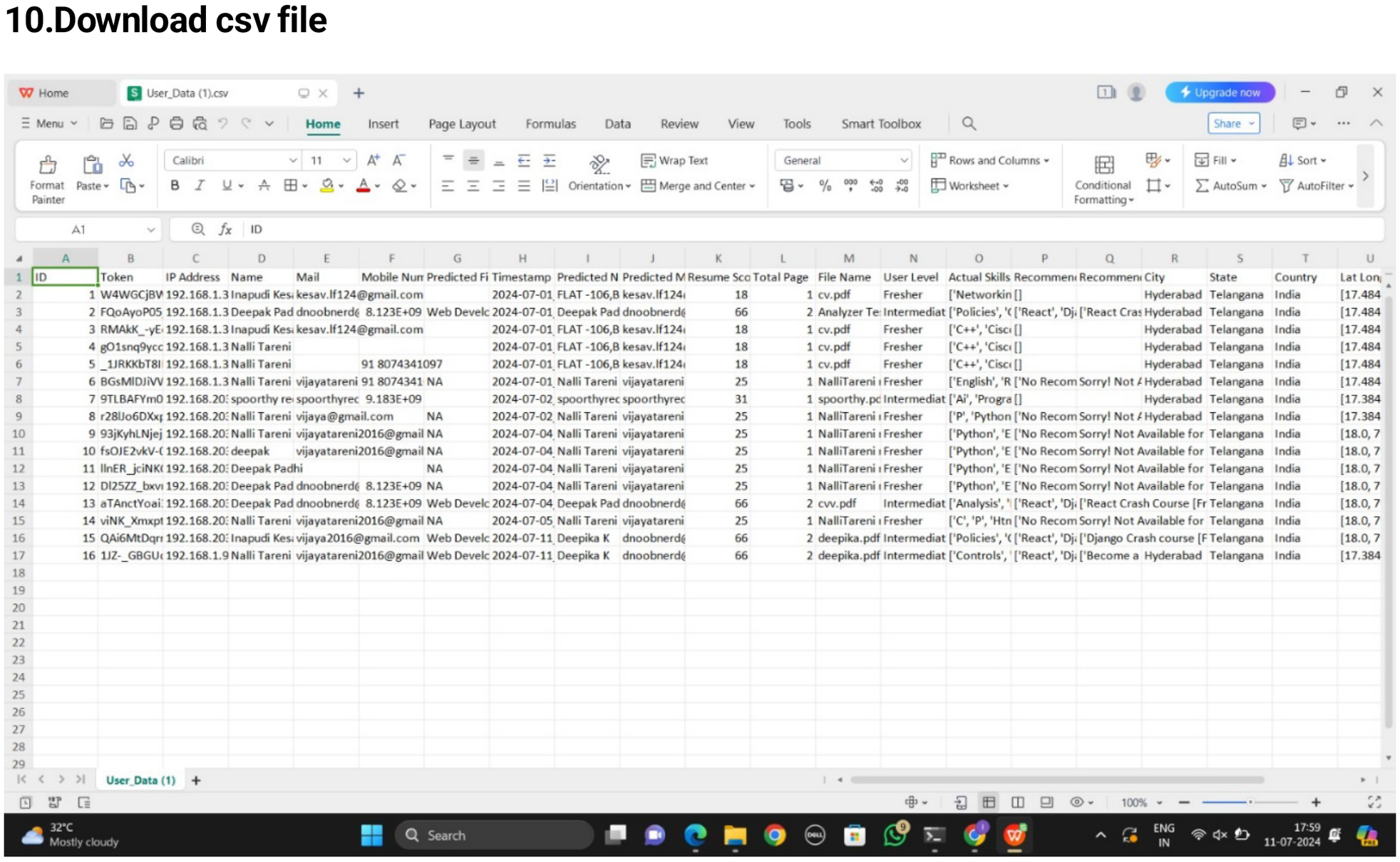


Fig 6.10

7.Conclusion & Future scope

7.1 Conclusion

An applicant cum recruiter-based Quick and easy to use Resume Analyzer. That analyze resume data and extract it into machine-readable output. Helps applicants with recommendations, prediction and analytics. Helps recruiter by automatically store, organize, and analyze resume data to find the best candidate. Can be widely used by any organization to analyze and get insights of a resume.

7.2 Future Scope

Add more fields for other roles, and its recommendations respectively. Ranking out the resume based on score and view individual user details. Decide more accurately and authentically, whether or not to offer candidate a job.

8.REFERENCES

1. https://www.udemy.com/course/the-full-stack-data-scientist-bootcamp/

2. https://docs.streamlit.io/

3. https://www.ijitee.org/wp-content/uploads/papers/v9i7/F4078049620.pdf

4. https://www.academia.edu/32543544/Resume_Parser_with_Natural_Language_Processing

5. https://www.rchilli.com/blog/resume-parsing-101/

6. https://en.wikipedia.org/wiki/R%C3%A9sum%C3%A9_parsing