# RESUME SCREENING


GUIDED BY
**Mrs. JYOTHI P JOY**
ASST PROFESSOR
DEPT OF CSE

| | |
|---|---|
| **1.AATHIRA P R** | **TJE19CS001** |
| **2.ABHIRAMI ANANTHAKUMAR** | **TJE19CS005** |
| **3.AMAL CEZNA** | **TJE19CS017** |
| **4.JEWELNA T J** | **TJE19CS045** |


# CONTENTS


- Introduction
- Existing System
- Literature Survey
- Problem Identification
- Proposed System
- System Architecture
- Modules
  - Upload Resume
  - Analyzer & Ranking
  - Questionnaire & Mail
- System Specifications
- Conclusion
- References

# INTRODUCTION

- Resume Screening is the primary step in the hiring process.
- Screening resumes is a time consuming but an important part of the selection process.
- Evaluates the candidates' resumes and determines whether they are qualified for a role.
- Manual screening - Consumes a lot of time and effort.

# Continuation…

- Resume Screening is the process of evaluating the candidates' resumes based on a specific requirement.

- The basic Job Hiring process consists of 5 steps:-
  - Resume shortlisting.
  - Technical written test.
  - Technical, behavioral, managerial interviews.
  - HR Interview.
  - Offering the job.

# EXISTING SYSTEM

- Manual Screening.

Problems in this Approach:

1. Time Consuming.
2. Recruiters are under a lot of pressure.
3. Unnecessary resource allocation.
4. Inefficient.
5. Errors.

# LITERATURE SURVEY

| PAPER NAME | PROBLEMS | SOLUTION | DRAWBACK |
|---|---|---|---|
| Manual Screening (support only pdf). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure. | Every resume is checked individually. | Due to many resumes and little time available for processing. |
| Manual screening in web (pdf). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure. | Resume is checked with the help of web helper. | can be miss out<br>Can create bug. |

| PAPER NAME | PROBLEMS | SOLUTION | DRAWBACK |
|---|---|---|---|
| Screening with automatic scoring (pdf). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure. | They analyze based on scores (education qualification). | They will miss out the skilled peoples. |
| Screening with the help of cloud database(pdf). | Time consuming<br>Errors<br>Lot of pressure. | Storing the datas on the cloud so that easily can sort it out. | Can form bug due to heavy use of internet. |

| PAPER NAME | PROBLEMS | SOLUTION | DRAWBACK |
|---|---|---|---|
| Screening based on experience(pdf). | Time consuming<br>Inefficient<br>Lot of pressure. | They analyze based on scores (experience). | They will miss out the skilled peoples. |
| Aws method of screening process (Amazon) (pdf). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure. | Amazon web service cloud method. | only focus aws data. |

| PAPER NAME | PROBLEMS | SOLUTION | DRAWBACK |
|---|---|---|---|
| Resume Ideal Software with chatbot. | Time consuming<br>Errors<br>Lot of pressure and doubts. | Chatbot to engage with the talent pool. | Costly to use. |
| Different file format detecting using deep learning including size of file. | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure.<br>Only pdf files.. | Can upload different types of files with max size 500 MB. | Cannot upload more than 500 MB of file. |

| PAPER NAME | PROBLEMS | SOLUTION | DRAWBACK |
|---|---|---|---|
| Resume parsing with search function methodology (pdf). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure. | All the data is converted to a structured format for quick sorting and searching. | They will miss out the skilled peoples.. |
| Newton Software or Paycor Recruiting<br><br>(Docx, RTF, PDF, and more). | Time consuming<br>Errors<br>Inefficient<br>Lot of pressure.<br>Only pdf files. | Support for different popular formats like Docx, RTF, PDF, and more. | Took a long time to detect. |

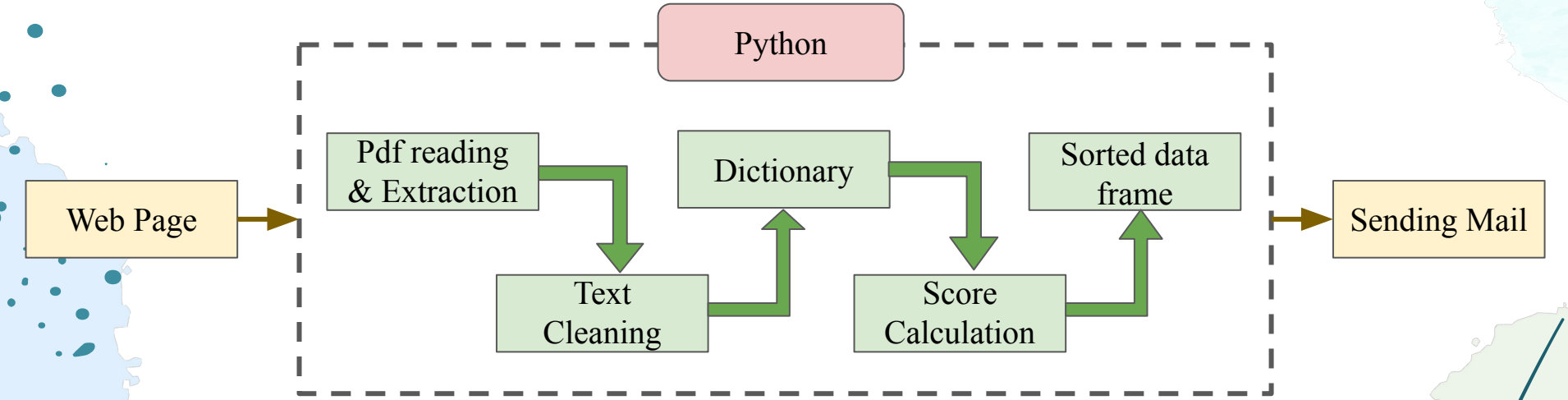
# PROBLEM IDENTIFICATION

- Companies often receive lots of resumes for each job posting.
- Hiring the right talent is a challenge for all businesses.
- The biggest challenge in resume screening is volume.
- Time-consuming to go through all the resumes that are submitted.

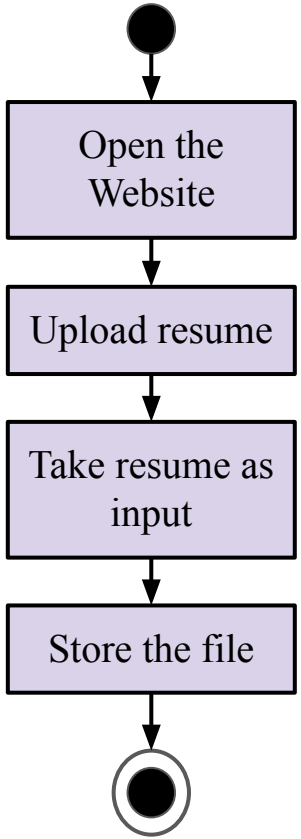
# PROPOSED SYSTEM

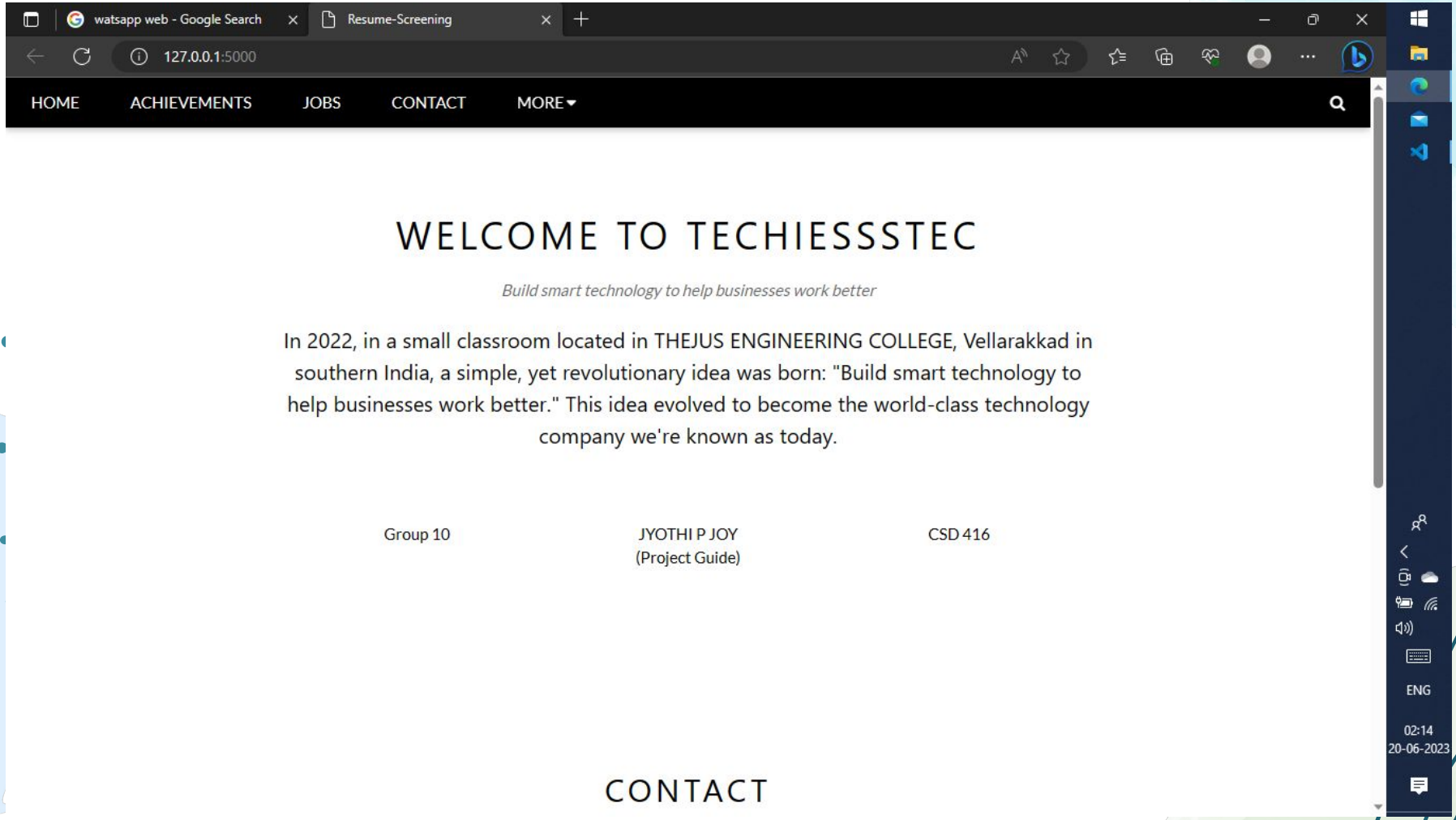
- Selecting the resumes with the required credentials.
- Select the resumes with the desired skills.
- Selecting the resumes customized for the job.
- Checking the applicant's information.

# SYSTEM ARCHITECTURE



# MODULE 1- UPLOAD RESUME




watsapp web - Google Search
Resume-Screening
127.0.0.1:5000
HOME
ACHIEVEMENTS
JOBS
CONTACT
MORE
WELCOME TO TECHIESSSTEC
Build smart technology to help businesses work better
In 2022, in a small classroom located in THEJUS ENGINEERING COLLEGE, Vellarakkad in southern India, a simple, yet revolutionary idea was born: "Build smart technology to help businesses work better." This idea evolved to become the world-class technology company we're known as today.
Group 10
JYOTHI P JOY
(Project Guide)
CSD 416
CONTACT
ENG
02:14
20-06-2023

```
<div class="w3-bar w3-black w3-card">
  <a class="w3-bar-item w3-button w3-padding-large w3-hide-medium w3-hide-large w3-right" href="javascript:void(0)" onclick="myFunctic
  <a href="" class="w3-bar-item w3-button w3-padding-large">HOME</a>
  <a href="#band" class="w3-bar-item w3-button w3-padding-large w3-hide-small">ACHIEVEMENTS</a>
  <a href="/file" class="w3-bar-item w3-button w3-padding-large w3-hide-small">JOBS</a>
  <a href="#contact" class="w3-bar-item w3-button w3-padding-large w3-hide-small">CONTACT</a>
  <div class="w3-dropdown-hover w3-hide-small">
    <button class="w3-padding-large w3-button" title="More">MORE <i class="fa fa-caret-down"></i></button>
    <div class="w3-dropdown-content w3-bar-block w3-card-4">
      <a href="#" class="w3-bar-item w3-button">MAPS</a>
      <a href="#" class="w3-bar-item w3-button">SERVICE</a>
      <a href="#" class="w3-bar-item w3-button">HELP</a>
    </div>
  </div>
  <a href="javascript:void(0)" class="w3-padding-large w3-hover-red w3-hide-small w3-right"><i class="fa fa-search"></i></a>
</div>
```

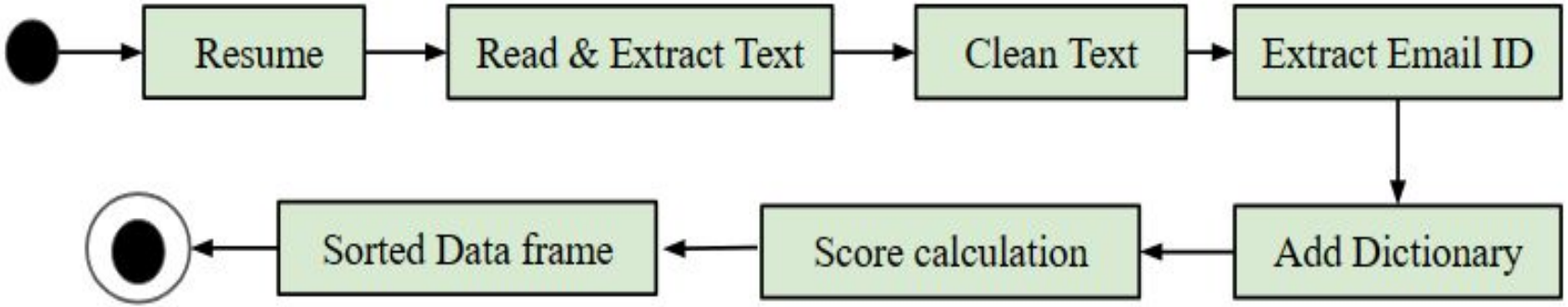
## SCREENING RESUME

Upload PDF

Choose file Browse

Upload

# MODULE 2
# ANALYZING & RANKING



# PDF file opening, reading and text extraction

```
210        pdfFileObj = open('uploads/{}'.format(name), 'rb')
211        pdfReader = PyPDF2.PdfReader(pdfFileObj)
212        num_pages = len(pdfReader.pages)
213        count = 0
214        text = ""
215        while count < num_pages:
216            pageObj = pdfReader.pages[count]
217            count += 1
218            text += pageObj.extract_text()
```

# Text cleaning

```
220        def cleanResume(resumeText):
221            resumeText = re.sub('http\S+\s*', ' ', resumeText)
222            resumeText = re.sub('RT|cc', ' ', resumeText)
223            resumeText = re.sub('#\S+', '', resumeText)
224            resumeText = re.sub('@\S+', '  ', resumeText)
225            resumeText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ',
226                                resumeText)
227            resumeText = re.sub(r'[^\x00-\x7f]', r' ', resumeText)
228            resumeText = re.sub('\s+', ' ', resumeText)
229            return resumeText.lower()
```

# Extracting Email id from PDF

```
37
38  def extract_email(email_content):
39      pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b'
40      email_pattern_compile = re.compile(pattern)
41      email_extracted_result = email_pattern_compile.search(email_content)
42      email_extracted = email_extracted_result.group()
43      return email_extracted
44
```

# Dictionary

```
bidang = {
    'Project Management': ['administration', 'agile', 'feasibility analysis', 'finance', 'leader', 'leadership',
                           'management', 'milestones', 'planning', 'project', 'risk management', 'schedule',
                           'stakeholders', 'teamwork', 'communication', 'organization', 'research',
                           'public speaking', 'problem solving', 'negotiation', 'team management',
                           'time management', 'adaptability', 'policy knowledge', 'reporting', 'technical',
                           'motivation'],

    'Backend': ['flask', 'laravel', 'django', 'ruby on rails', 'express.js', 'codeigniter', 'golang', 'mysql',
                'postgres', 'mongodb', 'relational database', 'non relational database', 'nosql',
                'application programming interface', 'object oriented programming'],

    'Frontend': ['react', 'angular', 'vue.js', 'svelte', 'jquery', 'backbone.js ', 'ember.js', 'semantic-ui',
                 'html', 'css', 'bootstrap', 'javascript', 'jquery', 'xml', 'dom manipulation', 'json'],

    'Data Science': ['math', 'statistic', 'probability', 'preprocessing', 'machine learning', 'data visualization',
                     'python', 'r programming', 'tableau', 'natural language processing', 'data modeling',
                     'big data', 'deep learning', 'relational database management', 'clustering', 'data mining',
                     'text mining', 'jupyter', 'neural networks', 'deep neural network', 'pandas', 'scipy',
                     'matplotlib', 'numpy', 'tensorflow', 'scikit learn', 'data analysis', 'data privacy',
                     'enterprise resource planning', 'oracle', 'sybase', 'decision making', 'microsoft excel',
                     'data collection', 'data cleaning', 'pattern recognition', 'google analytics'],

    'Devops': ['networking', 'tcp' 'udp', 'microsoft azure', 'amazon web services', 'alibaba cloud', 'google cloud',
               'docker', 'kubernetes', 'virtual machine', 'cloud computing', 'security', 'linux', 'ubuntu',
               'debian', 'arch linux', 'kali linux', 'automation', 'containers', 'operations', 'security',
               'testing', 'troubleshooting']
}
```

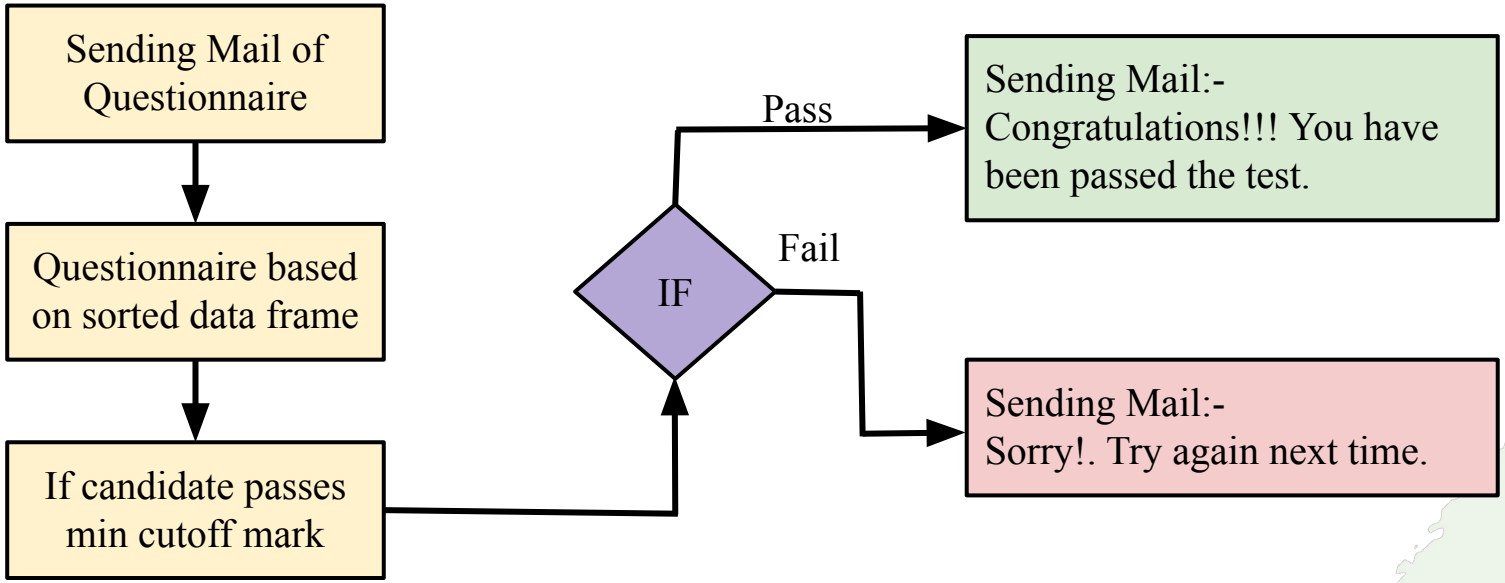
# Scores calculation per area

```
151         scores = []
152
153         for area in bidang.keys():
154             if area == 'Project Management':
155                 for word_project in bidang['Project Management']:
156                     if word_project in text:
157                         project += 1
158                         project_list.append(word_project)
159                 scores.append(project)
160
161             elif area == 'Backend':
162                 for word_backend in bidang['Backend']:
163                     if word_backend in text:
164                         backend += 1
165                         backend_list.append(word_backend)
166                 scores.append(backend)
167
168             elif area == 'Frontend':
169                 for word_frontend in bidang['Frontend']:
170                     if word_frontend in text:
171                         frontend += 1
172                         frontend_list.append(word_frontend)
173                 scores.append(frontend)
174
175             elif area == 'Data Science':
176                 for word_data in bidang['Data Science']:
177                     if word_data in text:
178                         data += 1
179                         data_list.append(word_data)
180                 scores.append(data)
181
```

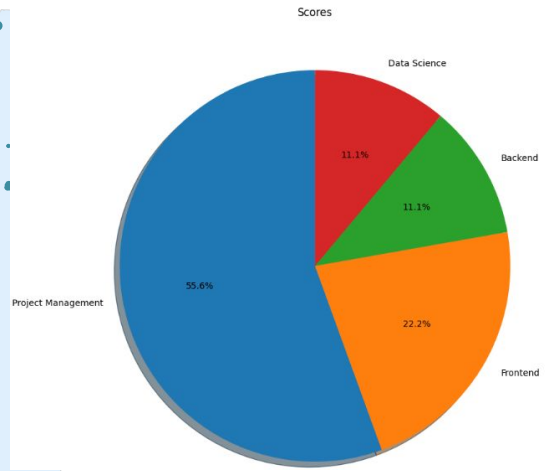
# Sorted data frame for final scores creation and Pie chart creation

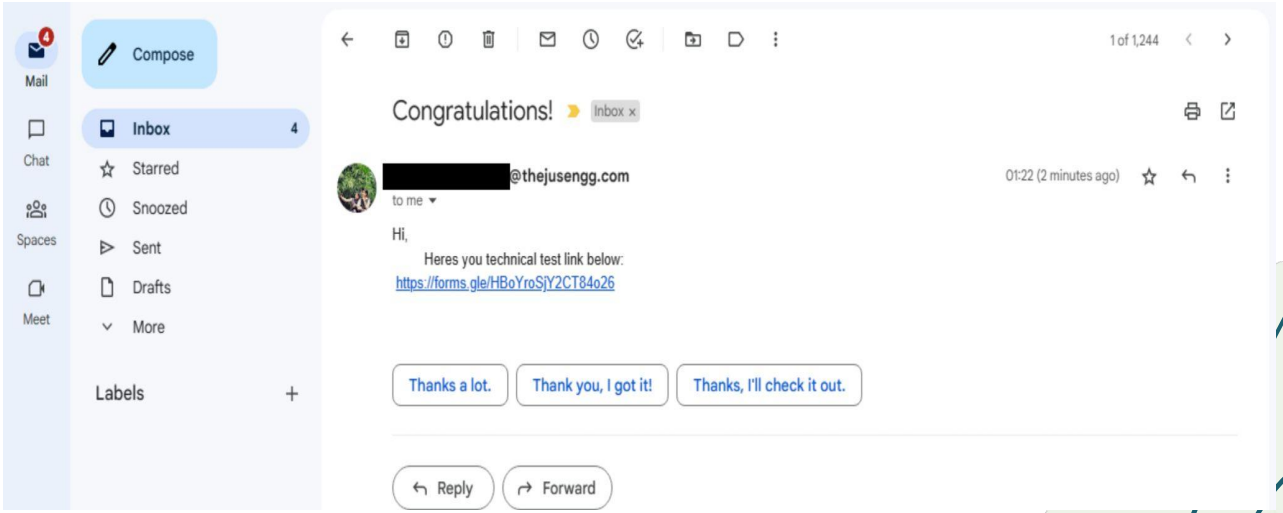
```
318     summary = pd.DataFrame(scores, index=bidang.keys(), columns=['score']).sort_values(by='score', ascending=False).loc[
319         lambda df: df['score'] > 0]
320
321     fig, ax = plt.subplots(figsize=(10, 10))
322     ax.pie(summary['score'], labels=summary.index, autopct='%1.1f%%', startangle=90, shadow=True)
323     ax.set_aspect('equal')
324     ax.set_title("Scores")
325     buf = BytesIO()
326     ax.figure.savefig(buf, format="png")
327     data = base64.b64encode(buf.getbuffer()).decode("ascii")
```

# MODULE 3
# QUESTIONNAIRE & MAIL



```
328
329     if summary['score']['Project Management'] > 0:
330         subject="Congratulations!"
331         msg = Message(subject, sender="aathiraprcse2019@thejusengg.com", recipients=[extracted_email])
332         msg.body = "Hi, \n\t Heres you technical test link below: \n https://forms.gle/HBoYroSjY2CT84o26 "
333         mail.send(msg)
334
335
```



# SYSTEM SPECIFICATIONS

System requirements:-

- Website Creation.
- Languages used : Python, HTML, CSS.
- Framework - Flask.
- IDE - Vscode.

# CONCLUSION

- Resume Screening is one of the most critical steps in the recruitment cycle.
- Very easy for recruiters.
- CVs can be screened easily and efficaciously.

# REFERENCES


- https://towardsdatascience.com/resume-screening-with-python-1dea360be49b
- https://testdriven.io/courses/learn-flask/intro/
- https://phoenixnap.com/kb/install-flask
- https://flask.palletsprojects.com/en/2.2.x/quickstart/

# THANK YOU!