ML based Career Service (MBCS)
INFO 7390 ADS Project Proposal

April 27, 2018

Team :Alpha Beta Gamma

Sreerag Mandakathil Sreenath
001838559
mandakathil.s@husky.neu.edu

Shreya Chudasama
001828562
chudasama.s@husky.neu.edu

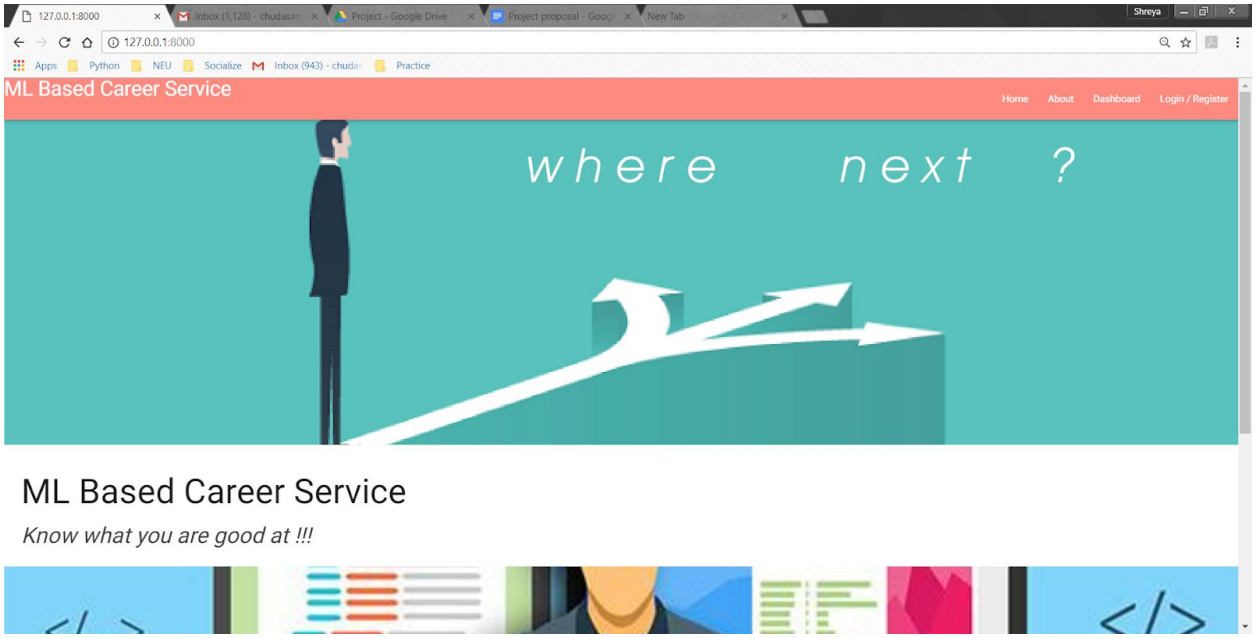
Aahana Khajanchi
001824402
khajanchi.a@husky.neu.edu

Github link : https://github.com/sreeragsreenath/ADS-Final-Project

Web App URL : http://ec2-52-32-210-114.us-west-2.compute.amazonaws.com:8000/

Project Proposal

MBC is Web based platform for candidates who are seeking a job and recruiter who are looking for the right talent. The platform scrapes and analyzes publicly available job posting data from sites like Glassdoor, LinkedIn, etc. The system will recommend the candidate which job title will fit him/her based on their resume. The system will also give job links to the application form. The platform will also help us to gain statistical information about the latest job trend.



Stakeholders

The Supervising authority

The Supervising authority is a person that governs the entire system. Having supervisory powers over some aspects of management decision-making. Basically a person with more power than others.

The Beneficiaries (or users)

The beneficiaries (or users) are all natural persons or corporate body, students , recipients of the services provided by MBCS.

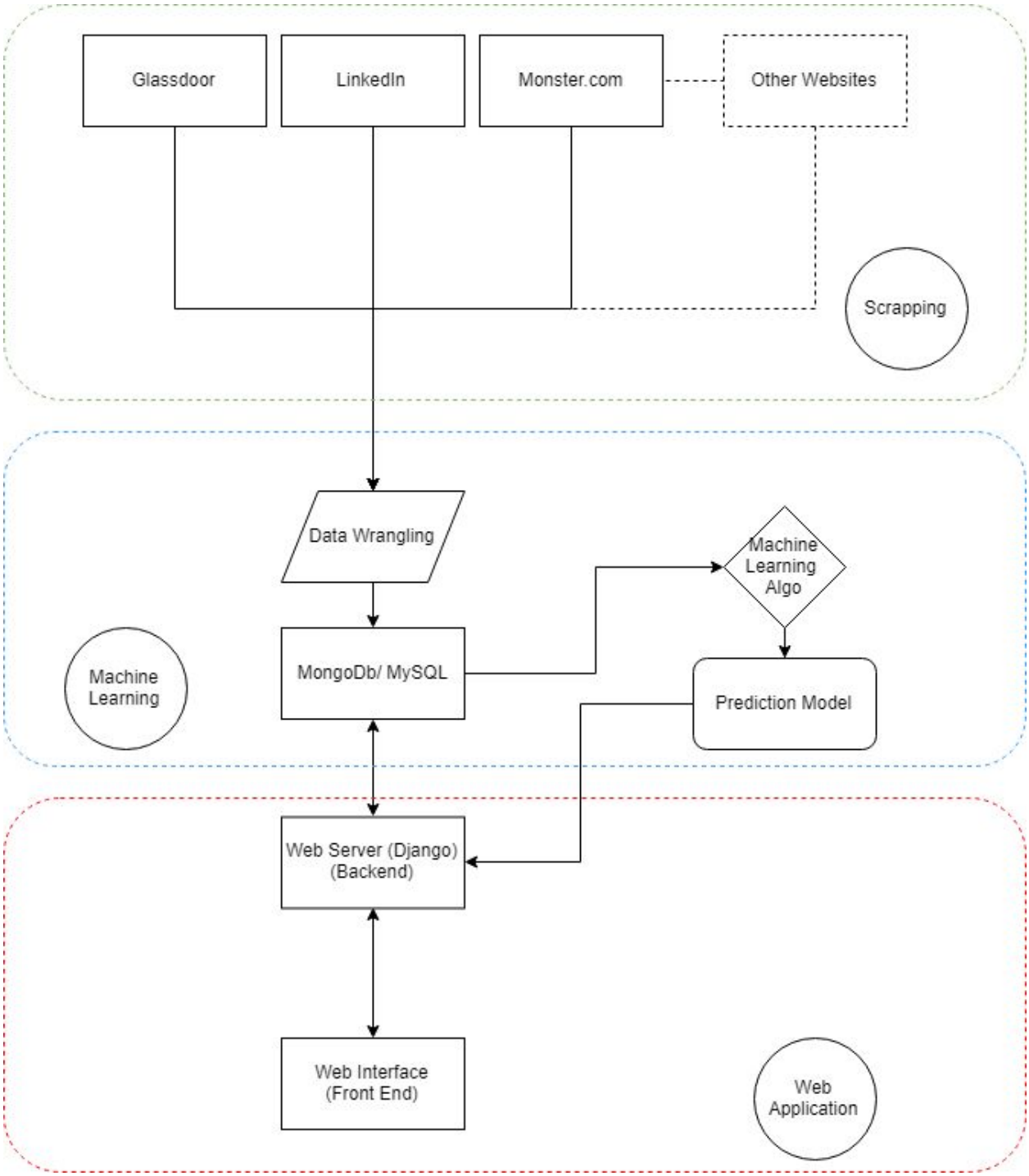
The Recruiters

A person whose job is to enlist or enroll people as employees as members of an organization.

The steps involved

1. We have used an existing dataset provided by Kaggle
   DataSet :
2. **Scrapped** information and skill sets from websites such as **Indeed**
   a. This will be done by giving the url as input to a web scraper
   b. Library such as **scrapy** will be used to run multiple web crawlers on a single website to optimise scrapping process
3. Analysing the data and segregating the content to skills and tools mandatory and the expertise level
   a. The data is stored in **SQLite** database supported by **Django** server for further processing
4. We tried out different machine learning models and hybrid techniques to optimise the result
5. Made a web based application to interact with the system through which the user can upload his/her data.
   a. The **web based** application and the **authentication** is done using python **Django**.

6. The Web application allows the user to upload the document and it will Parse the users resume and scrape relevant information i.e. tools, skills, location, work experience and serve it as parameters for our prediction model.
7. After the model is executed, it will output "**Best suited position**" according to his/her resume along with **Ranking**. It also outputs **latest Job postings** where the user can apply.
8. The entire process is pipelined using ***LUIGI.***
9. We have deployed the Django application on **EC2** server and storing and retrieving the model from **Amazon S3**.
10. Finally we have **dockerized** the entire application, so a person with any operating system can simply pull the docker image and run it on his own machine.


Glassdoor
LinkedIn
Monster.com
Other Websites
Scrapping
Data Wrangling
Machine Learning Algo
Machine Learning
MongoDb/ MySQL
Prediction Model
Web Server (Django) (Backend)
Web Interface (Front End)
Web Application

Deployment Details

1. Luigi

   Luigi is a Python package that helps you build complex pipelines of batch jobs. It handles dependency resolution, workflow management, visualization, handling failures, command line integration, and much more.

   Luigi command

   python luigi_pipeline.py upload model to S3 --workers 2 --akey <Your_S3_ID>--skey <Your_S3_Key>

2. Docker

   Docker is an open source tool that automates the deployment of the application inside software container.

   Docker command

   docker build -f dockerfile . -t sreeragsreenath/finalads

   docker run -it -p 8000:8000 -p 8082:8082 sreeragsreenath/finalads

3. Amazon S3

   Amazon Simple Storage Service (Amazon S3) is a web service that provides highly scalable cloud storage. Amazon S3 provides easy to use object storage, with a simple web service interface to store and get any amount of data from anywhere on the web.

Python Library

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
We have built out **web application** using Django

PyPDF2

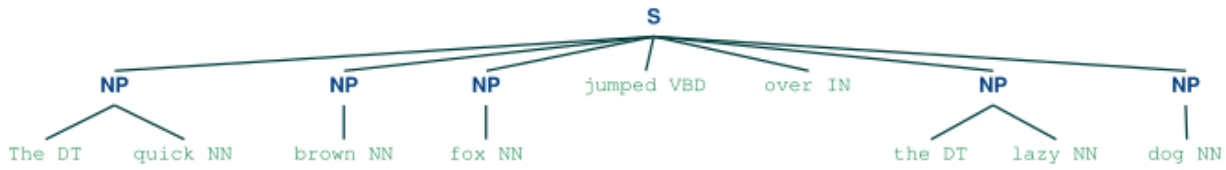
A Pure-Python library built as a PDF toolkit. It is capable of:

- extracting document information (title, author, …)
- splitting documents page by page
- merging documents page by page
- merging multiple pages into a single page
- encrypting and decrypting PDF files

We have used this library to **parse the Resume**.

NLTK (Natural Language Toolkit)

Is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) in the Python programming language. NLTK includes graphical demonstrations and sample data. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.



We have used NLTK to **tokenize** the words and make a **dictionary** of those words .

TextBlob

TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
We have used Textblob to

Boto3

Boto is the Amazon Web Services (AWS) SDK for Python, which allows us to write software that makes use of Amazon services like S3 and EC2. Boto provides an easy to use, object-oriented API as well as low-level direct service access.
We have used Boto3 to **upload and reload** the **model to S3** and **host the web application** on EC2 server.

Integrating Tableau

Tableau helps people transform data into actionable insights. Explore with limitless visual analytics. Build dashboards and perform ad hoc analyses in just a few clicks. Share your work with anyone and make an impact on your business. Tableau helps you see the stories in your data.
We have used Tableau to let our user understand the Jobs and the position frequency.

Methodology

We are using the dataset provided by Kagel(https://www.kaggle.com/madhab/jobposts) and scraping the data from Indeed.com

Below is data set which consist

jobpost – The original job post

date – Date it was posted in the group

Title – Job title

Company - employer

AnnouncementCode – Announcement code (some internal code, is usually missing)

Term – Full-Time, Part-time, etc

Eligibility -- Eligibility of the candidates

Audience --- Who can apply?

StartDate – Start date of work

Duration - Duration of the employment

Location – Employment location

JobDescription – Job Description

JobRequirment - Job requirements

RequiredQual -Required Qualification

Salary - Salary

ApplicationP – Application Procedure

OpeningDate – Opening date of the job announcement

Deadline – Deadline for the job announcement

Notes - Additional Notes

AboutC - About the company

Attach - Attachments

Year - Year of the announcement (derived from the field date)

Month - Month of the announcement (derived from the field date)

```
In [6]: data= pd.read_csv("./dataset/data_job_posts.csv")
        data
```

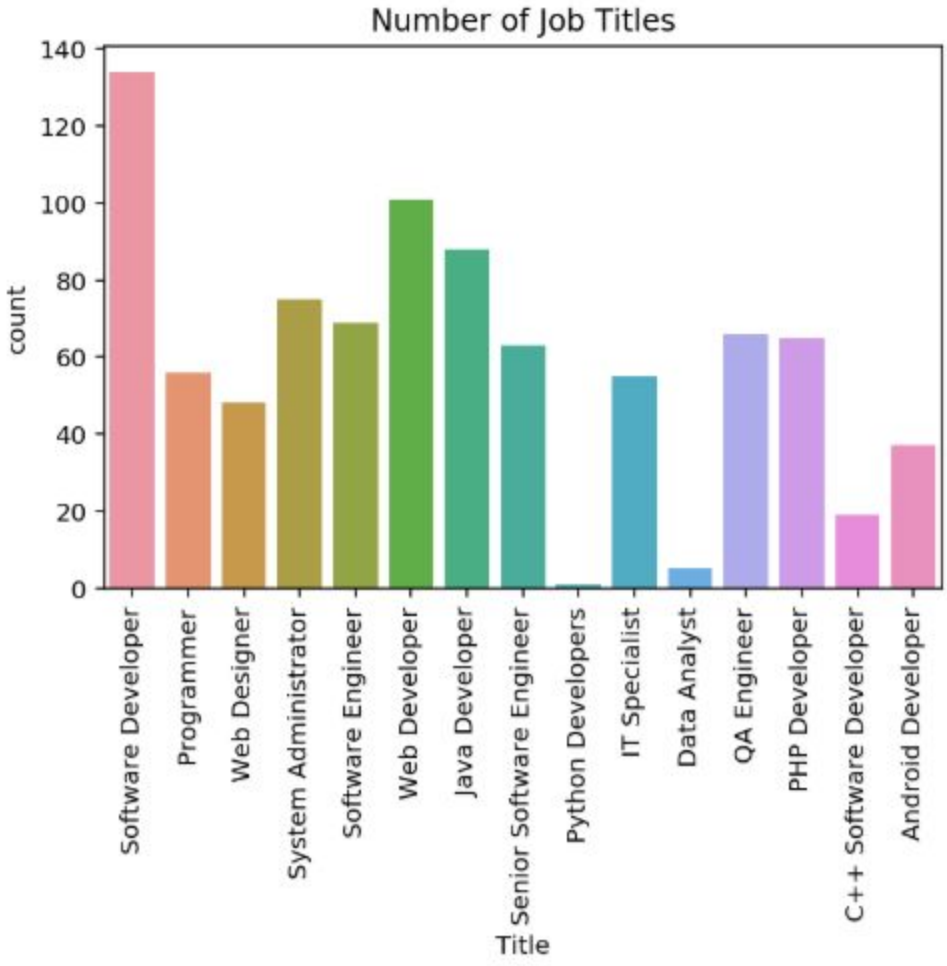
Out[6]:

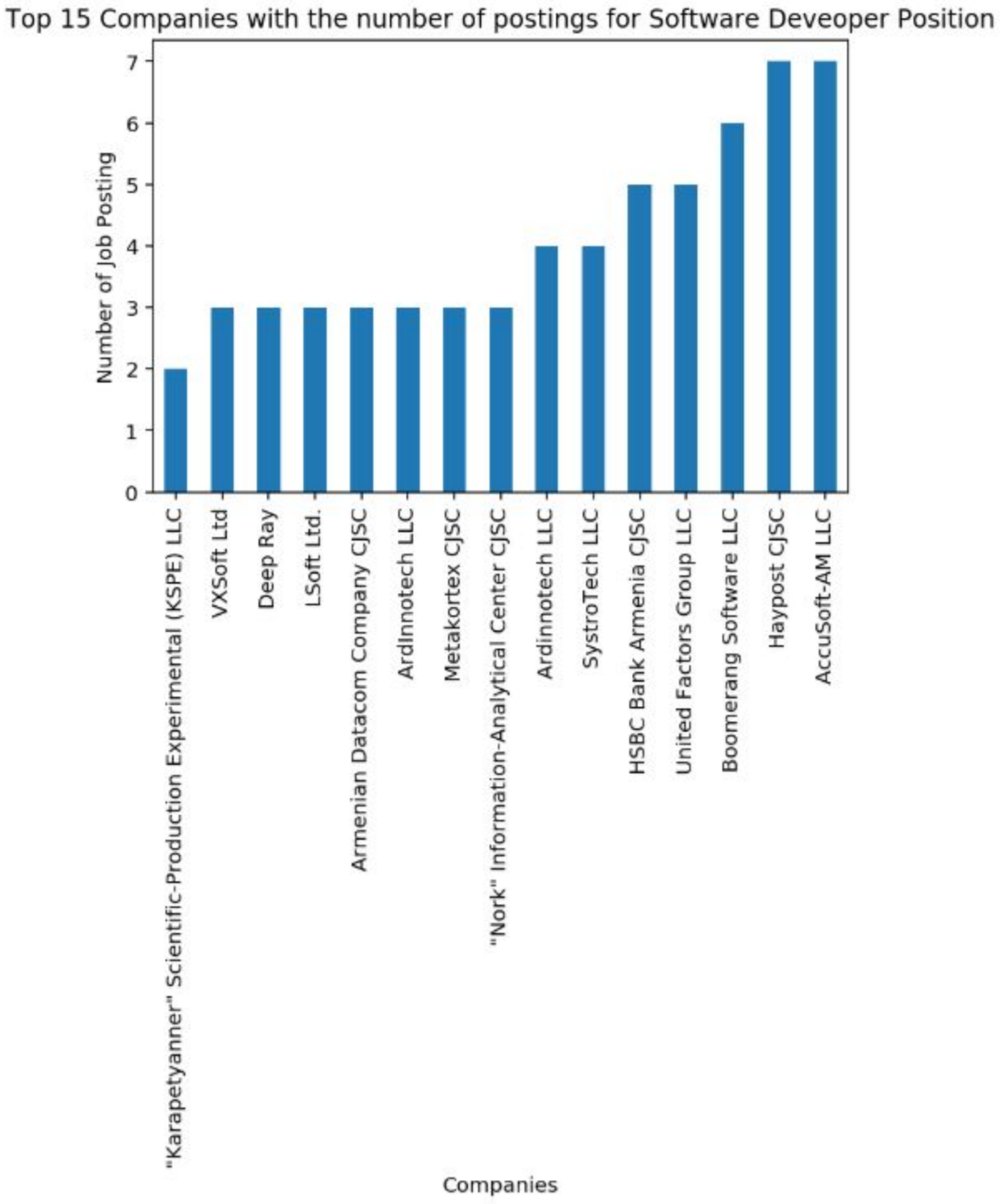
	jobpost	date	Title	Company	Announcer
0	AMERIA Investment Consulting Company\r\r\nJOB ...	Jan 5, 2004	Chief Financial Officer	AMERIA Investment Consulting Company	
1	International Research & Exchanges Board (IREX...	Jan 7, 2004	Full-time Community Connections Intern (paid i...	International Research & Exchanges Board (IREX)	
2	Caucasus Environmental NGO Network (CENN)\r\r\...	Jan 7, 2004	Country Coordinator	Caucasus Environmental NGO Network (CENN)	
3	Manoff Group\r\r\nJOB TITLE: BCC Specialist\r...	Jan 7, 2004	BCC Specialist	Manoff Group	
4	Yerevan Brandy Company\r\r\nJOB TITLE: Softwa...	Jan 10, 2004	Software Developer	Yerevan Brandy Company	

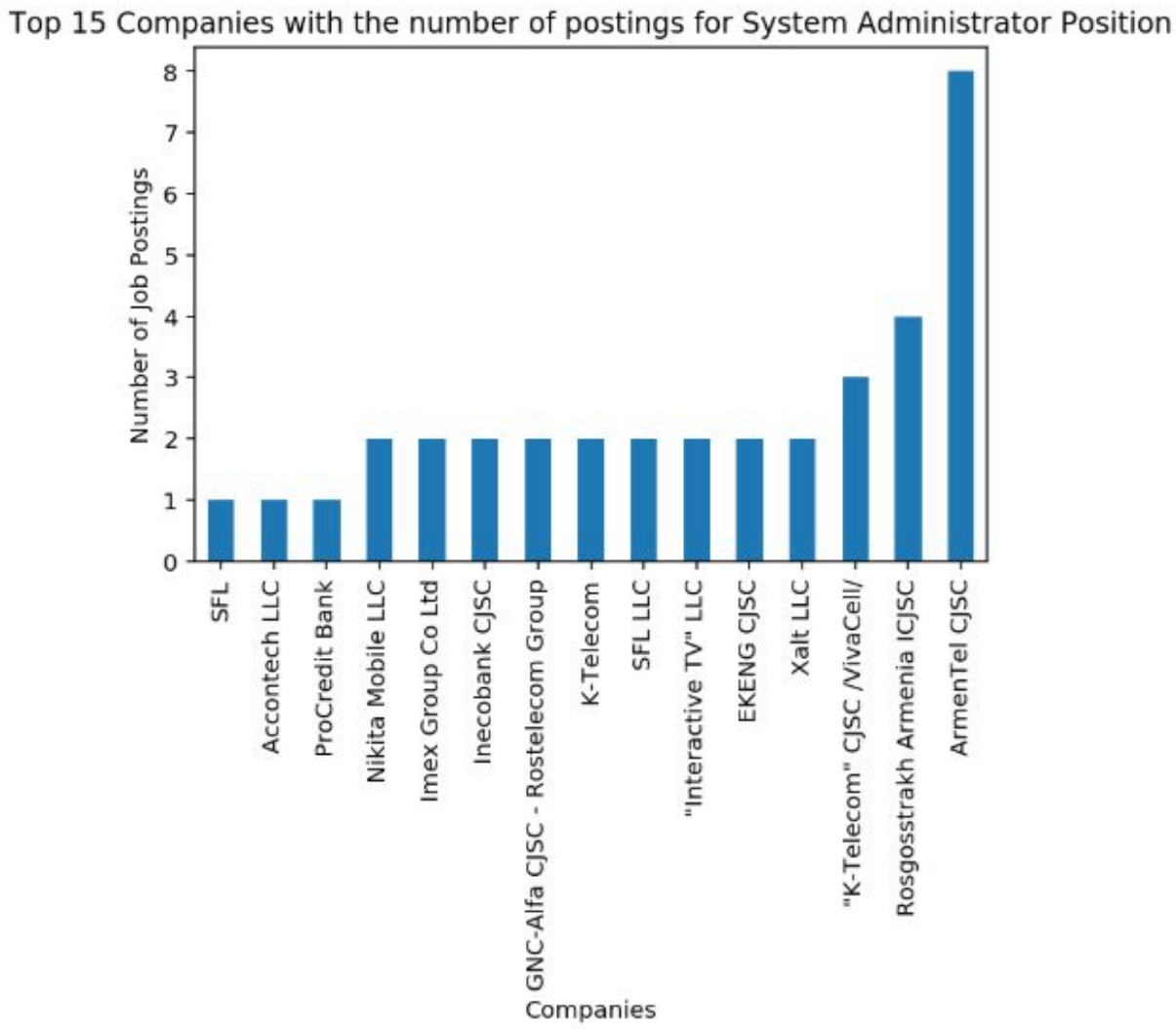
Exploratory Data Analysis

```
: import seaborn as sns
  sns.countplot(test2['Title'])
  plt.xticks(rotation = 'vertical')  # the x-axis labels
  plt.title("Number of Job Titles")
```

```
: Text(0.5,1,'Number of Job Titles')
```




Top 15 Companies with the number of postings for Software Deveoper Position
Number of Job Posting
0
1
2
3
4
5
6
7
"Karapetyanner" Scientific-Production Experimental (KSPE) LLC
VXSoft Ltd
Deep Ray
LSoft Ltd.
Armenian Datacom Company CJSC
ArdInnotech LLC
Metakortex CJSC
"Nork" Information-Analytical Center CJSC
Ardinnotech LLC
SystroTech LLC
HSBC Bank Armenia CJSC
United Factors Group LLC
Boomerang Software LLC
Haypost CJSC
AccuSoft-AM LLC
Companies



We are using **brown** and **punkt** package of nltk library and textblob to collect noun from the dataset

```
import nltk
nltk.download('brown')
nltk.download('punkt')
from textblob import TextBlob
```

```
train = test2[['jobpost','Title']]
```

```
def converttext(bag):
    blob = TextBlob(bag)
    blob.noun_phrases
    text = ' '.join(blob.noun_phrases)
    return text
```

```
train['jobpost'] = train['jobpost'].apply(lambda x: converttext(x))
```

```
train['jobpost']
```

```
4      yerevan brandy job title software developer po...
35     cit job title programmer position location yer...
92     synergy systems inc./armenia job title softwar...
122    acra credit job title web designer position lo...
196    boomerang sosftware llc title web designer loc...
```

Then we are storing each noun/word according to the position it belong as an array.

```
train.values
```

```
array([['yerevan brandy job title software developer position location yerevan armenia job responsibilities rendering technical
assistance database management systems realization sql servers maintenance activities participation software development projec
ts required qualifications university degree economical background excellent windows server networking tcp/ ip ms sql server vi
sual database software development good knowledge english remuneration will application procedures successful candidates cv rec
ommendation previous employers copy diploma relevant certificates color photo isakov yerevan armine.bibilyan @ ... resources ar
mine bibilyan please application letter job opportunity career url thanks application deadline january careercenter.am website
atmailbox @ ...',
        'Software Developer'],
       ['cit job title programmer position location yerevan armenia required qualifications work experience knowledge visual st
udio .net remuneration depends previous experience application procedures rosa karapetyan rosak @ ... abelyan yerevan 3-rd floo
r additional information short-listed please application letter job opportunity career url thanks application deadline february
careercenter.am website atmailbox @ ...',
        'Programmer'],
       ["synergy systems inc./armenia job title software developer position location yerevan armenia job description synergy sy
stems inc./armenia long-term position software developer core software development tasks synergy systems synergy main focus web
database web portal business intelligence knowledge management e-government solutions software developer quality software produ
ct commercial setting experience dynamic workplace solid software software development process job responsibilities specific ke
y responsibilities translate design requirements robust implementations technical aspects perform timely fashion perform qualit
y assurance tasks software products required qualifications degree computer information bachelor 's degree relevant field maste
```

Then we have used Naive Bayes classifier to classify the job title

```
from textblob.classifiers import NaiveBayesClassifier
```

```
cl = NaiveBayesClassifier(train.values)
```

Once our model is ready, we have pickel the model.

Pickling of the file

```
import pickle

object = cl
file = open('f2.obj','wb') #tried to use 'a' in place of 'w' ie. append
pickle.dump(object,file)
```

Below is the code where it parses the resume and then runs our model and outputs the result.

```
import PyPDF2    #PyPDF2 is a Pure-Python Library built as a PDF toolkit.
pdfFileObj = open('dataset/resume_shruti.pdf','rb')
pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
pdfReader.numPages
pageObj = pdfReader.getPage(0)
text = pageObj.extractText()    # Extracted text from pdf file
#------------------------------------------------
blob = TextBlob(text)
blob.noun_phrases
text = ' '.join(blob.noun_phrases)
text
#------------------------------------------------
print(cl.classify(text))
prob_dist = cl.prob_classify(text)
rank = []
for k in profiles:
    ar = [k,prob_dist.prob(k)]
    rank.append(ar)
rank = sorted(rank, key=lambda x: x[1],reverse=True)
rank
```

```
Software Engineer

[['Software Engineer', 0.9999985010408075],
 ['Programmer', 9.82483563816839e-07],
 ['Web Developer', 5.132524943535799e-07],
 ['Web Designer', 2.9506543278715137e-09],
 ['C++ Software Developer', 2.5989969461017965e-10],
 ['PHP Developer', 1.2611220875571593e-11],
 ['Senior Software Engineer', 7.517978005155277e-15],
 ['Android Developer', 1.4247193213301927e-16],
 ['Java Developer', 3.0084295771360356e-17],
```

We are even suggesting which jobs the user should apply.

```
['Joomag AM LLC', 1.0363434010959909e-35],
['Haypost CJSC', 5.134094240079218e-36],
['Synopsys Armenia CJSC', 1.3410876791699781e-36],
['Webb Fontaine Holding LLC', 4.0024973219915755e-40],
['Virtual Solution Global Services LLC', 3.4501690008497207e-44],
['AtTask', 1.938535525862578e-45],
['Lycos Armenia', 5.739296108823267e-47],
['Sourcio CJSC', 1.7841499575742133e-47],
['Monitis GFI CJSC', 1.0332234112206448e-48],
['Accept Employment Agency', 6.995266928082873e-53],
['VOLO LLC', 1.4648945993647089e-55],
['Macadamian AR CJSC', 2.1869971159910166e-57],
['EPAM Systems, Inc.', 2.7039895161890144e-59],
['Kubisys CJSC', 1.985079194649635e-59],
['SFL LLC', 4.8292733093243065e-62],
['AccuSoft-AM LLC', 8.20785985311596e-64],
['Metakortex CJSC', 4.789317924803619e-65],
['Converse Bank CJSC', 2.6256507919754177e-65],
['EpygiArm LLC', 1.2487966875542504e-68],
['Rosgosstrakh-Armenia ICJSC', 1.072267748883691e-68],
```

Scraping The Data from Indeed.com

So we are passing the Indeed URL to our scraper along with position and location and it parses all the job postings related to that position on Indeed.

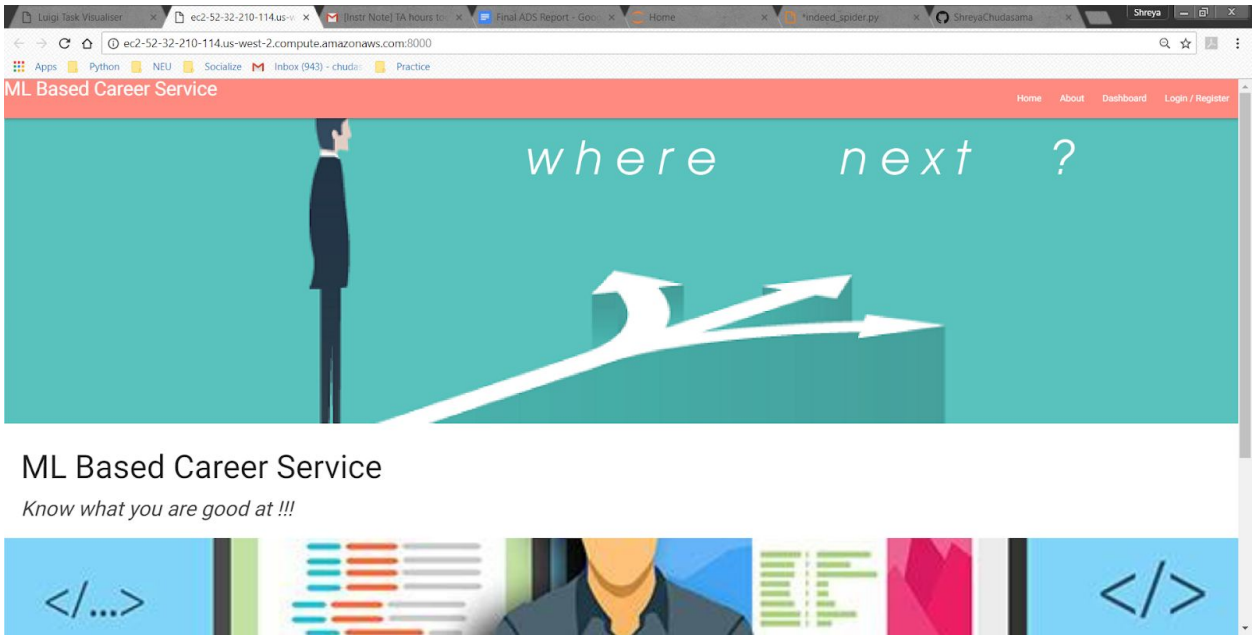
```
def start_requests(self):
    position = self.position
    location = self.location
    position_text = position
    position = position.replace(" ", "+")
    location = location.replace(" ", "%2C+")
    urls = [
        'https://www.indeed.com/jobs?q='+position+'&l='+location,
    ]
    for url in urls:
        yield scrapy.Request(url=url, callback=self.parse,  meta={'position': position_text})

def parse(self, response):
    position = response.meta['position']
    jobs = response.css("div.result")
    for job in jobs:
        # print(job.css("a.jobtitle::attr(href)").extract())
        next_page = job.css("a.jobtitle::attr(href)").extract_first()
        next_page = response.urljoin(next_page)
        print(next_page)
        # print(next_page)
        yield scrapy.Request(url = next_page, callback=self.parsejob, meta={'position': position, 'job_page' : next_page})
```

The parse job scrapes the data needed and stores it in a JSON file as a key value attribute.

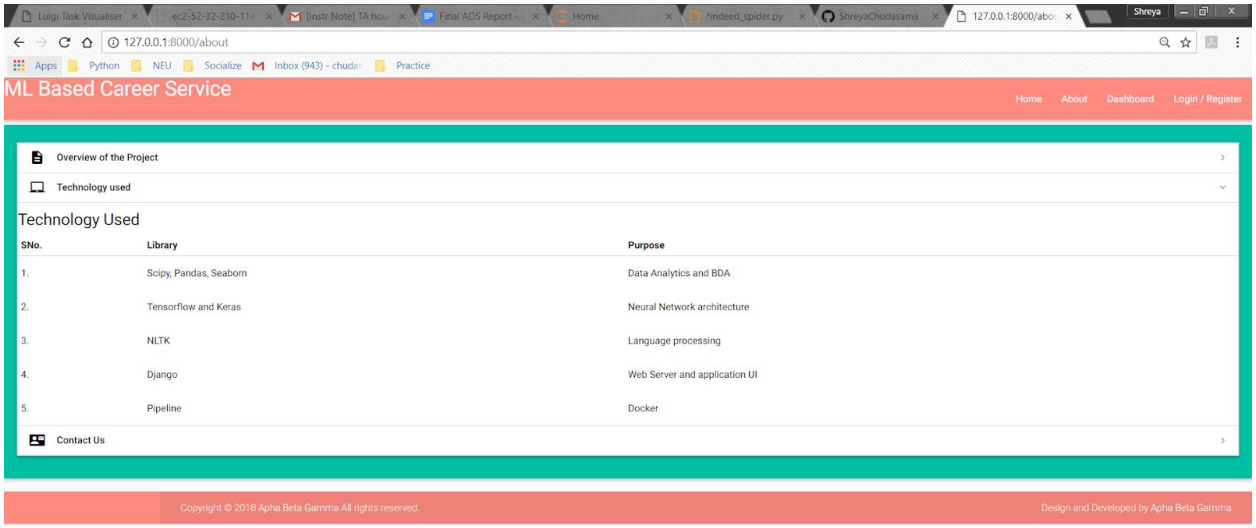
```
def parsejob(self, response):
    job_spec = response.css('title::text').extract()
    job_title = response.xpath("//b[@class = 'jobtitle']/font/text()").extract()
    job_desc = response.xpath("//table[@id = 'job-content']").css('td span').extract()
    company = re.sub("<.*?>", " ", str(job_desc[0]))
    job_desc = ''.join(job_desc)
    job_desc = re.sub("<.*?>", " ", str(job_desc))
    yield {
        'job_spec': job_spec,
        'job_title': job_title,
        'job_desc' : job_desc,
        'position' : response.meta['position'],
        'job_page' : response.meta['job_page'],
        'company' : company
    }
```

Web Application using Django

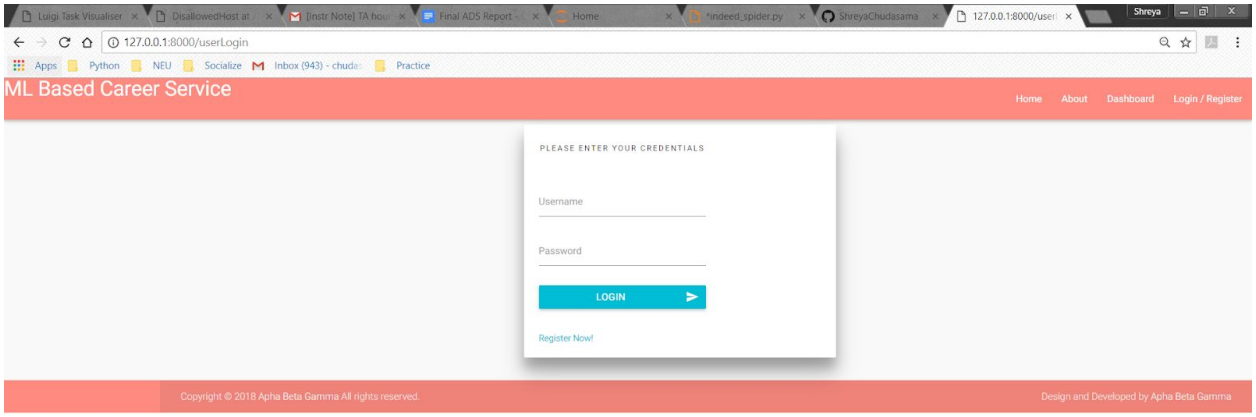
Home Page



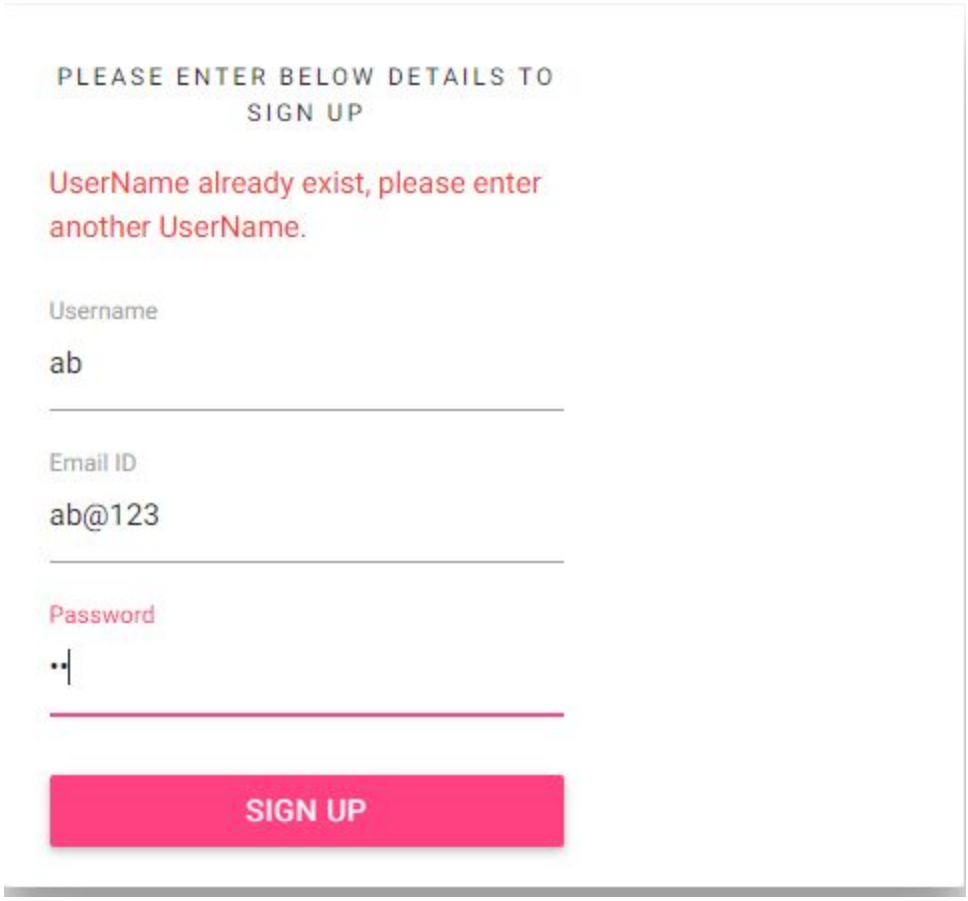
About Page



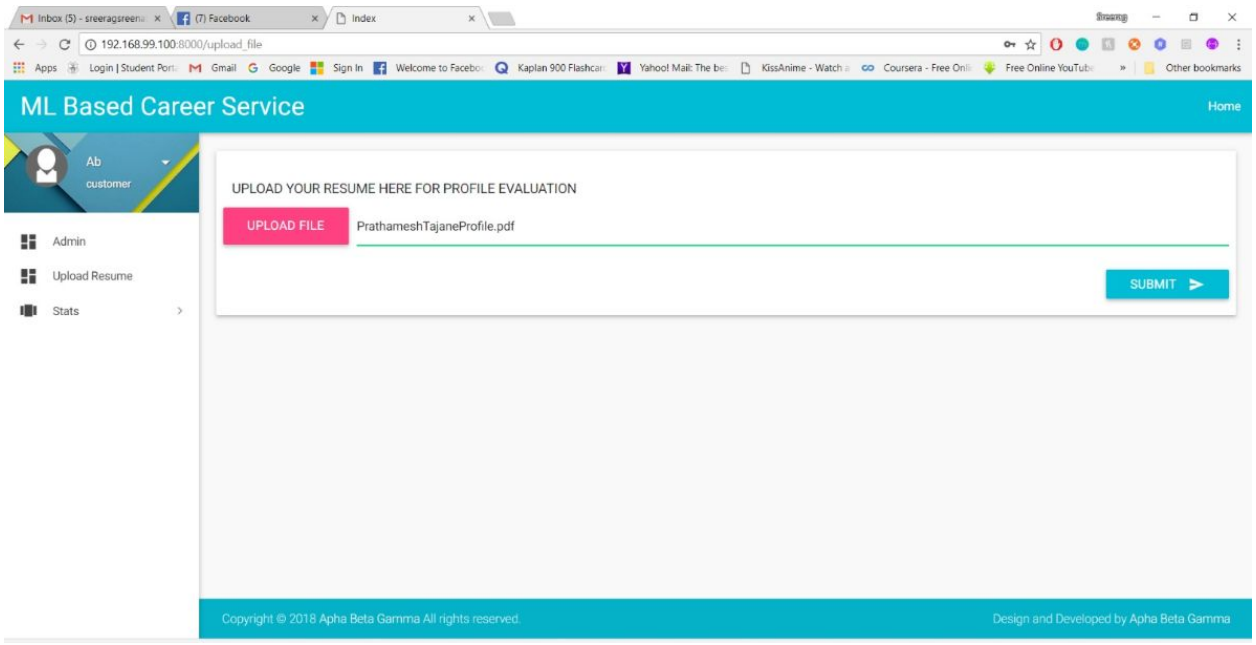
Login Page



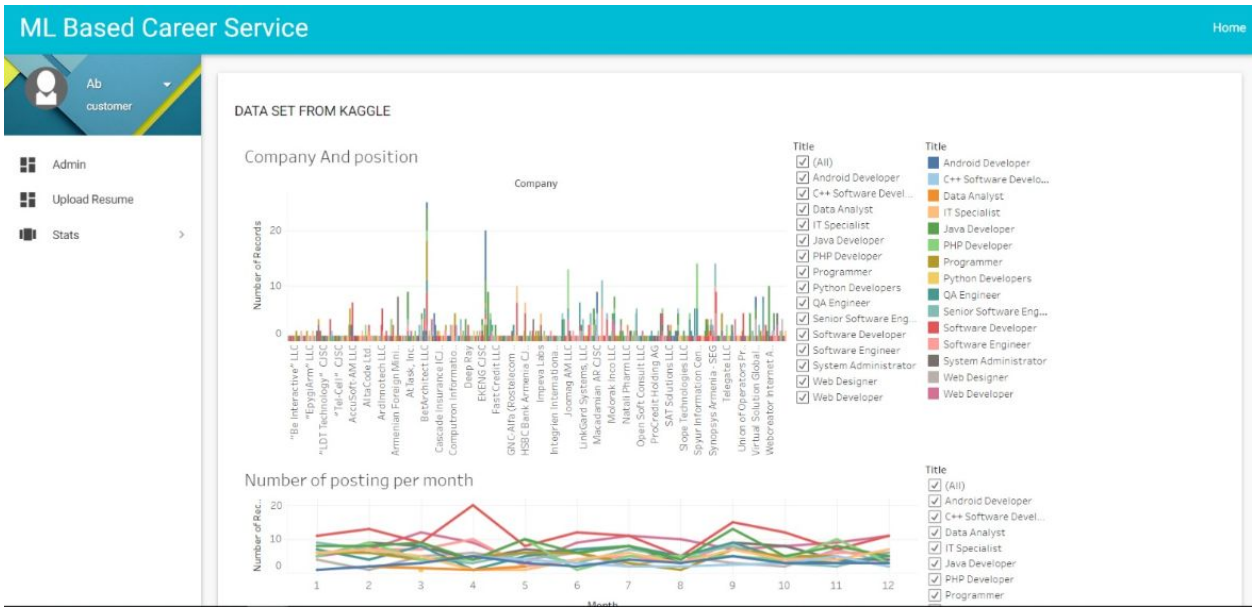
Validation for existing user on the register page

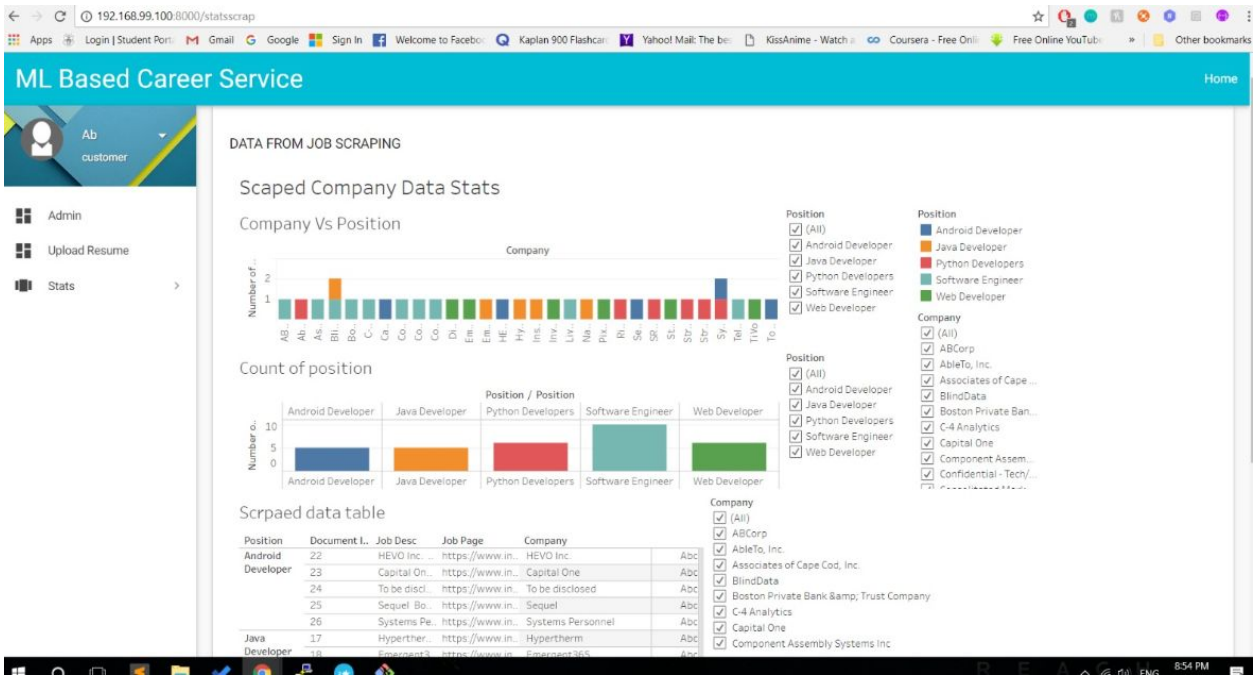

PLEASE ENTER BELOW DETAILS TO SIGN UP
UserName already exist, please enter another UserName.
Username
ab
Email ID
ab@123
Password
SIGN UP

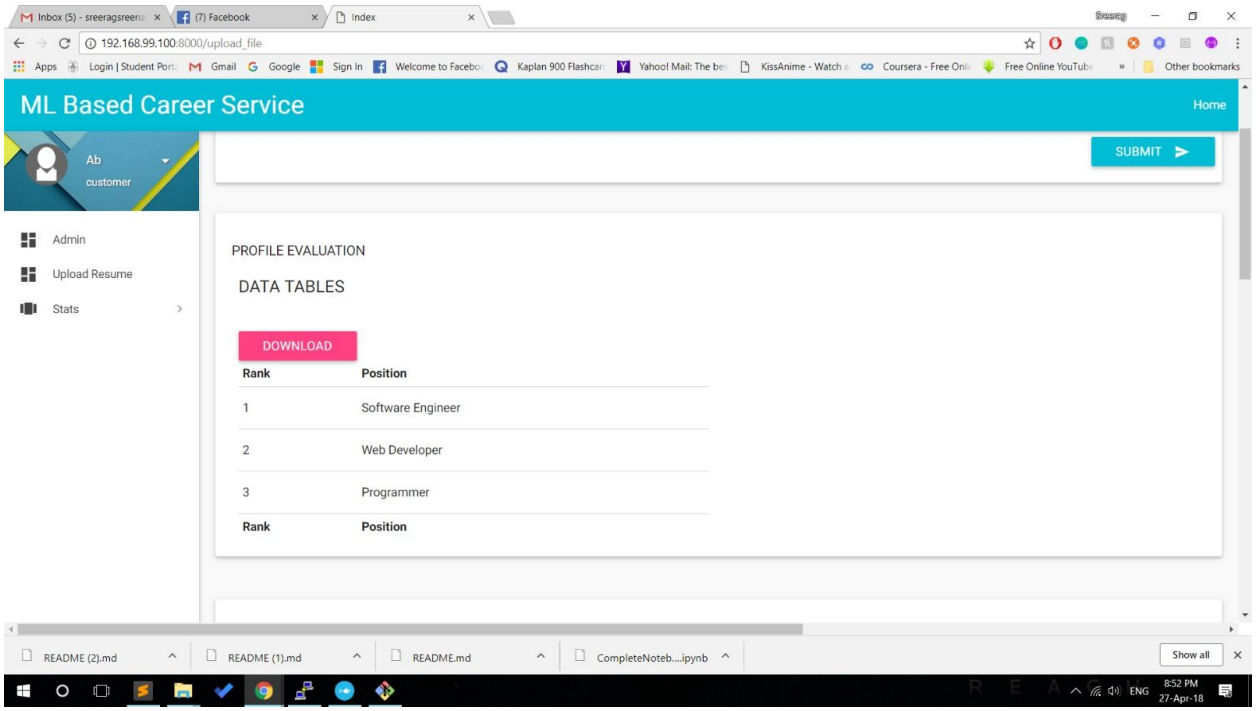
Dashboard



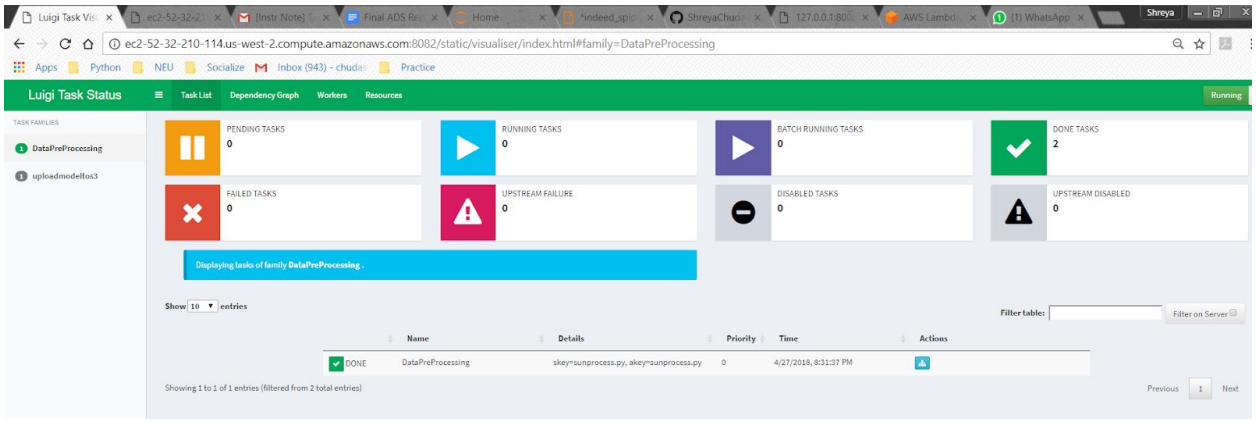
Statistics using Tableau







Luigi deamon



Implementation of Lamda

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running.

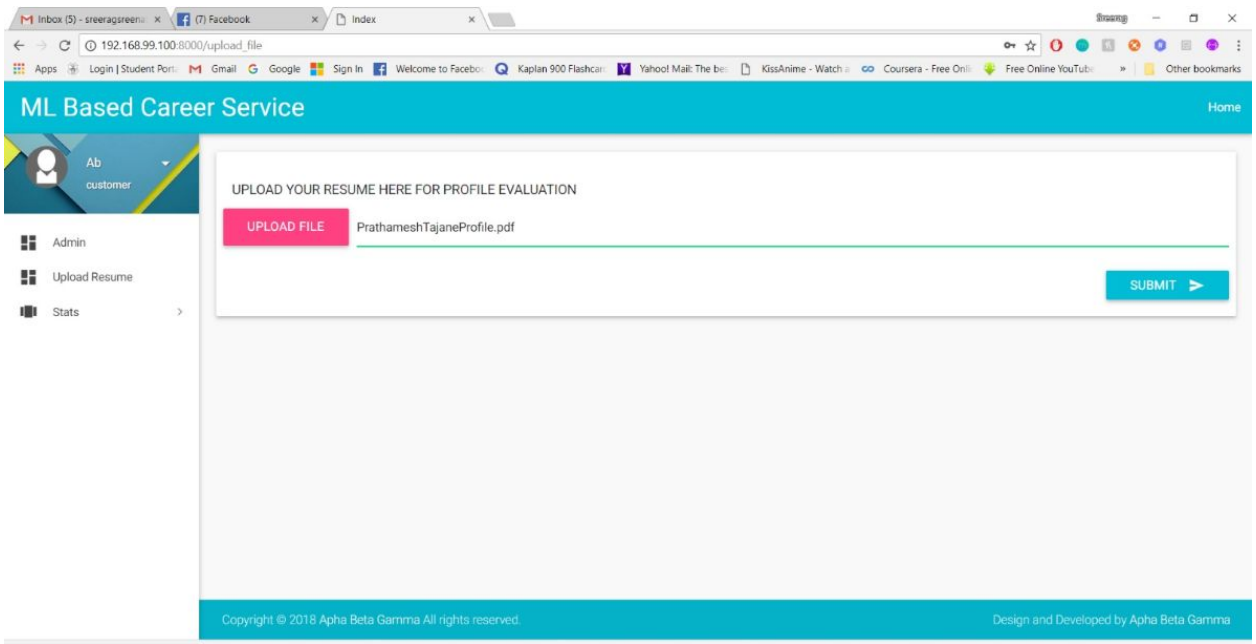
With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

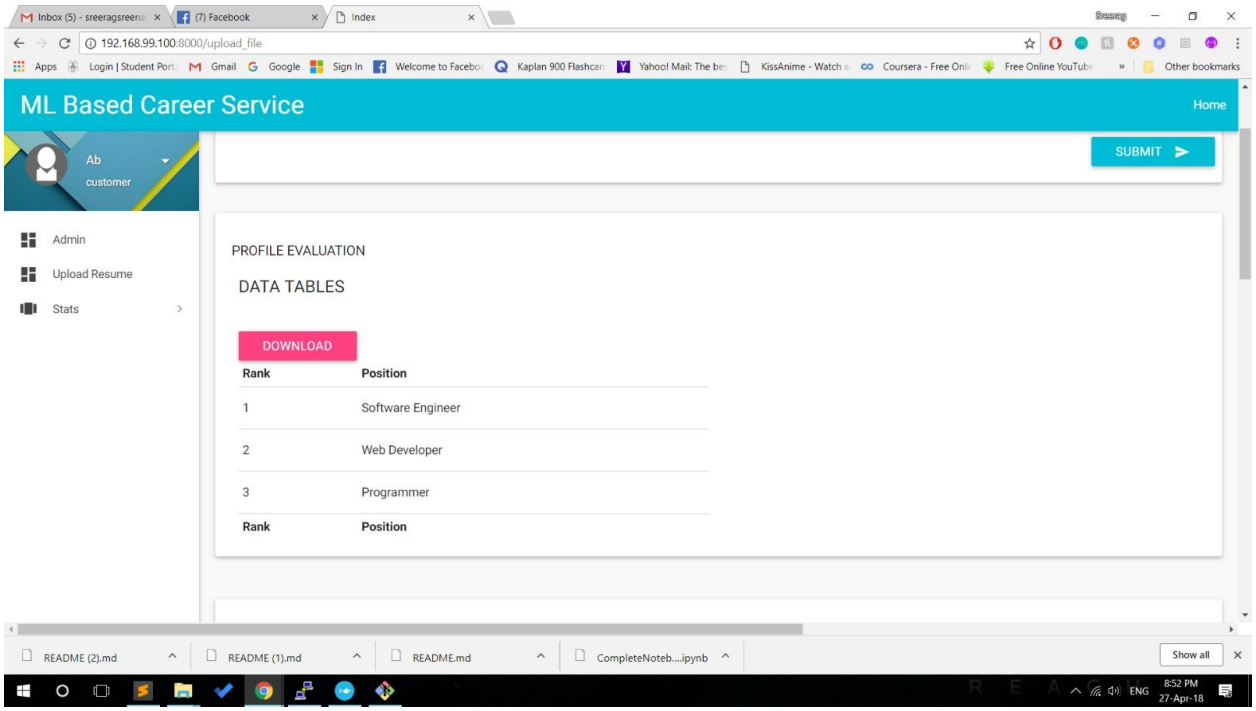
Code :

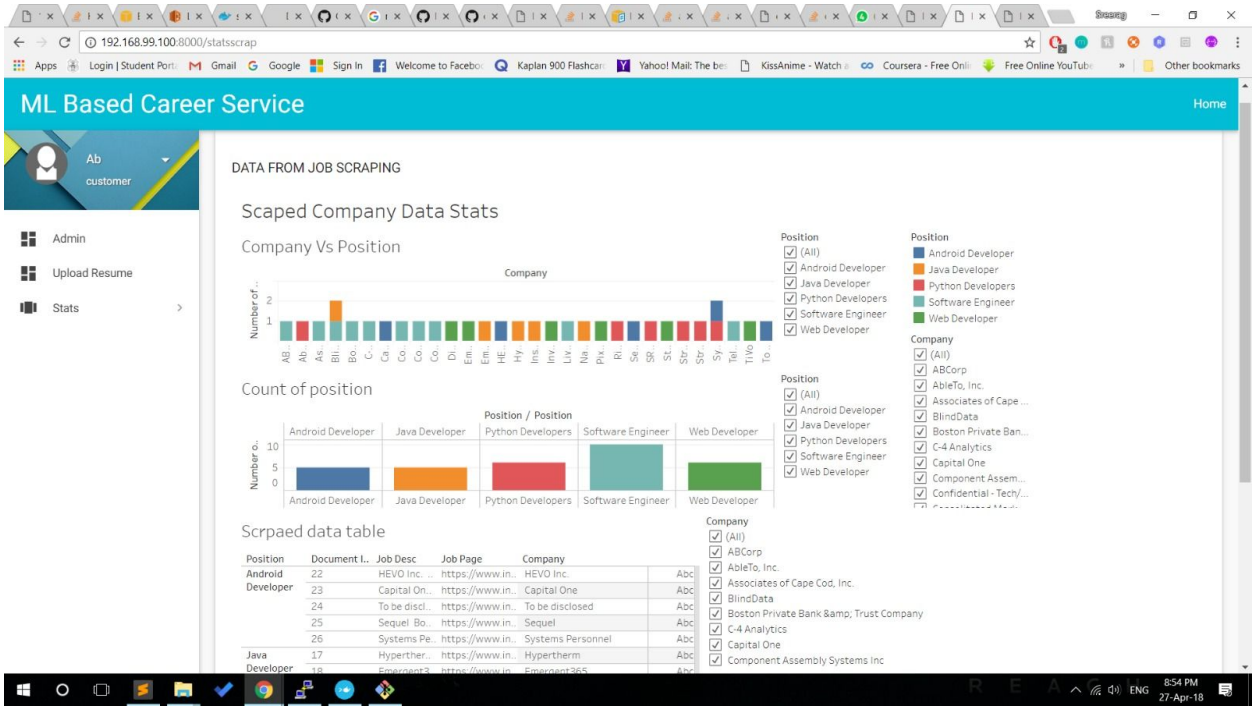
```
from botocore.vendored import requests

def lambda_handler(event, context):
   response = requests.get("http://ec2-52-41-199-219.us-west-
2.compute.amazonaws.com:8000/recompute")
   print(response.content)
   # TODO implement
   print("Triggered!!")
   return 'Hello from Lambda'
```

Conclusion

Our website will show you the Preferred Job Postings as well as the Positions one should apply.




ML Based Career Service
Home
Ab
customer
SUBMIT
Admin
Upload Resume
Stats
PROFILE EVALUATION
DATA TABLES
DOWNLOAD
Rank
Position
1
Software Engineer
2
Web Developer
3
Programmer
Rank
Position
README (2).md
README (1).md
README.md
CompleteNoteb....ipynb
Show all


ML Based Career Service
Home
Ab
customer
Admin
Upload Resume
Stats
DATA FROM JOB SCRAPING
Scaped Company Data Stats
Company Vs Position
Company
Number of ...
Count of position
Position / Position
Android Developer
Java Developer
Python Developers
Software Engineer
Web Developer
Number o...
Scrpaed data table
Position
Document I...
Job Desc
Job Page
Company
Android Developer
Java Developer
HEVO Inc.
Capital One
To be disclosed
Sequel
Systems Personnel
Hypertherm
Boston Private Bank & Trust Company
C-4 Analytics
Component Assembly Systems Inc

Advantages

1. The user will understand if his/her resume is corresponding to the Jobs that he is looking for and give him a clear picture. It will provide job links, related to the positions suggested.
2. User just have to upload their resume and everything else would be handled by the system,user won't have to write down the skills.

References

- https://luigi.readthedocs.io/en/stable/
- http://boto3.readthedocs.io/en/latest/guide/s3-examples.html
- http://boto.cloudhackers.com/en/latest/s3_tut.html
- https://towardsdatascience.com/beginners-guide-to-data-science-python-docker-3181fd321a5c
- https://doc.scrapy.org/en/latest/intro/tutorial.html