**INTRODUCTION**

* 1. **OVERVIEW**

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. College recruiting is typically a tactic for medium- to large-sized companies with high-volume recruiting needs, but can range from small efforts (like working with university career centers to source potential candidates) to large-scale operations (like visiting a wide array of colleges and attending recruiting events throughout the spring and fall semester).Campus recruitment often involves working with university career services centers and attending career fairs to meet in-person with college students and recent graduates.

Our solution revolves around the placement season of a Business School in India. Where it has various factors on candidates getting hired such as work experience, exam percentage etc., Finally it contains the status of recruitment and remuneration details. We will be using algorithms such as KNN, SVM and ANN. We will train and test the data with these algorithms. From this the best model is selected and saved in.pkl format. We will be doing flask integration and IBM deployment.

* 1. **PURPOSE**

According to statistics 1.6 million students pass from CS department every year. The demand for skilled and qualified students is rising day by day. Thus, the company use a good amount of capital in recruiting students from in-campus and off-campus because number of skilled and qualified students are very low. Colleges and Institution needs to focus on practical knowledge of real world rather than completing their syllabus.

Placements are the biggest opportunities in the life of a student and they need to be fully prepared while attempting it. Placement Predictor system helps in predicting whether a student will get placement or not. This system can also be helpful for identifying the areas where student needs to work on for placement. This system uses student’s details like academic marks, coding skills, etc. This system uses previous year placement statistics and student dataset for the placement prediction so the placement cell of the organization could set up a placement anticipated rundown for the present students.

**PROBLEM DEFINITION AND DESIGN THINKING**

**2.1 EMPATHY MAP**

An empathy map is a square divided into four quadrants with the user or client in the middle. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels, and does.

WHY IS THIS SO HARD?

I WAS EXPECTING SOMETHING DIFFERENT

WHAT IS BEST FOR ME?

WHERE SHOULD I START?

**THINKS**

**SAYS**

**FEELS**

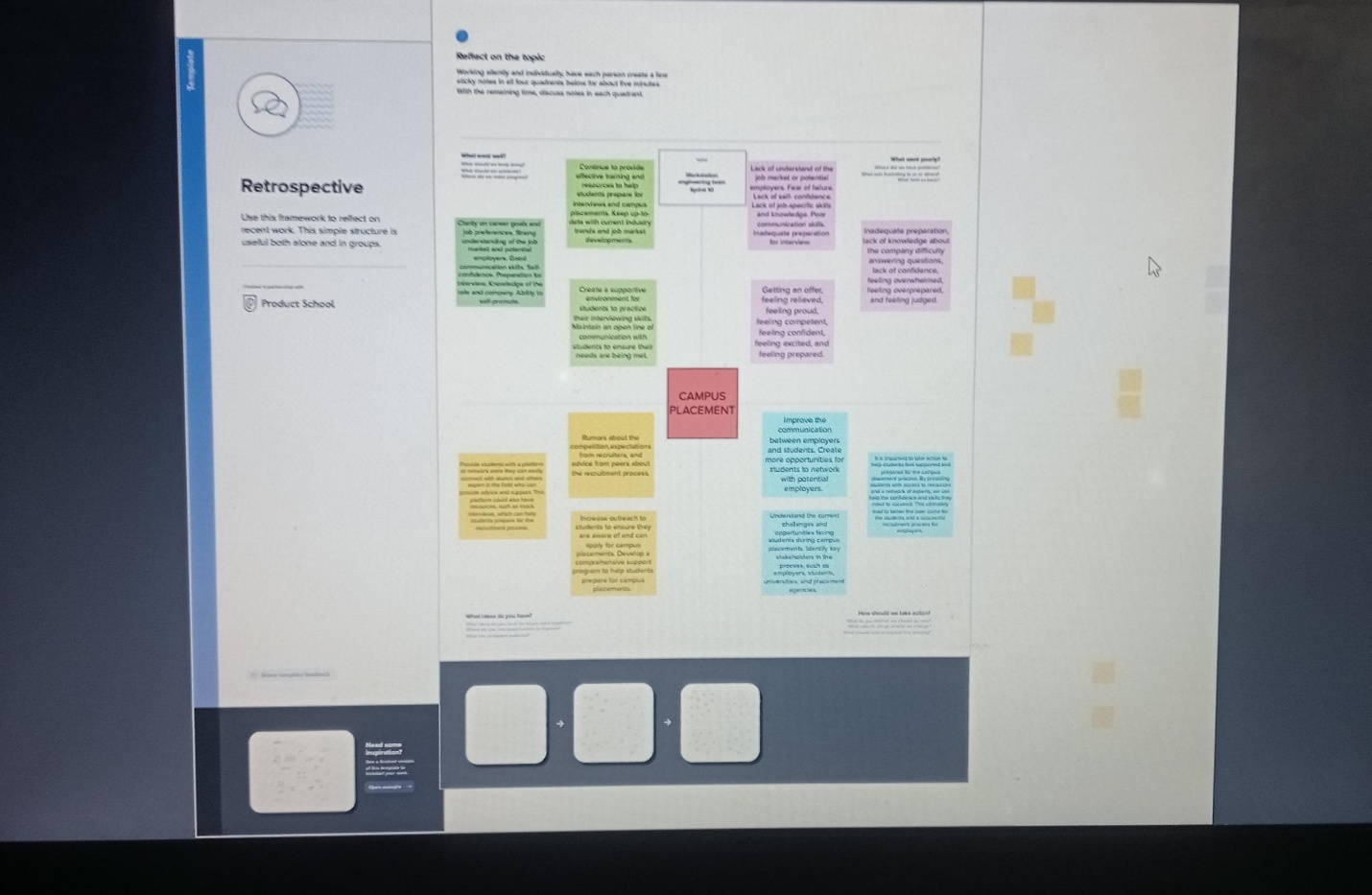
**DOES**

EXCITED

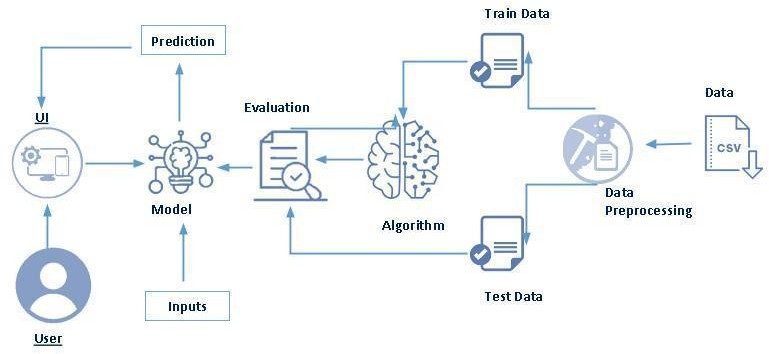
LIST THE PRONS\ CONS

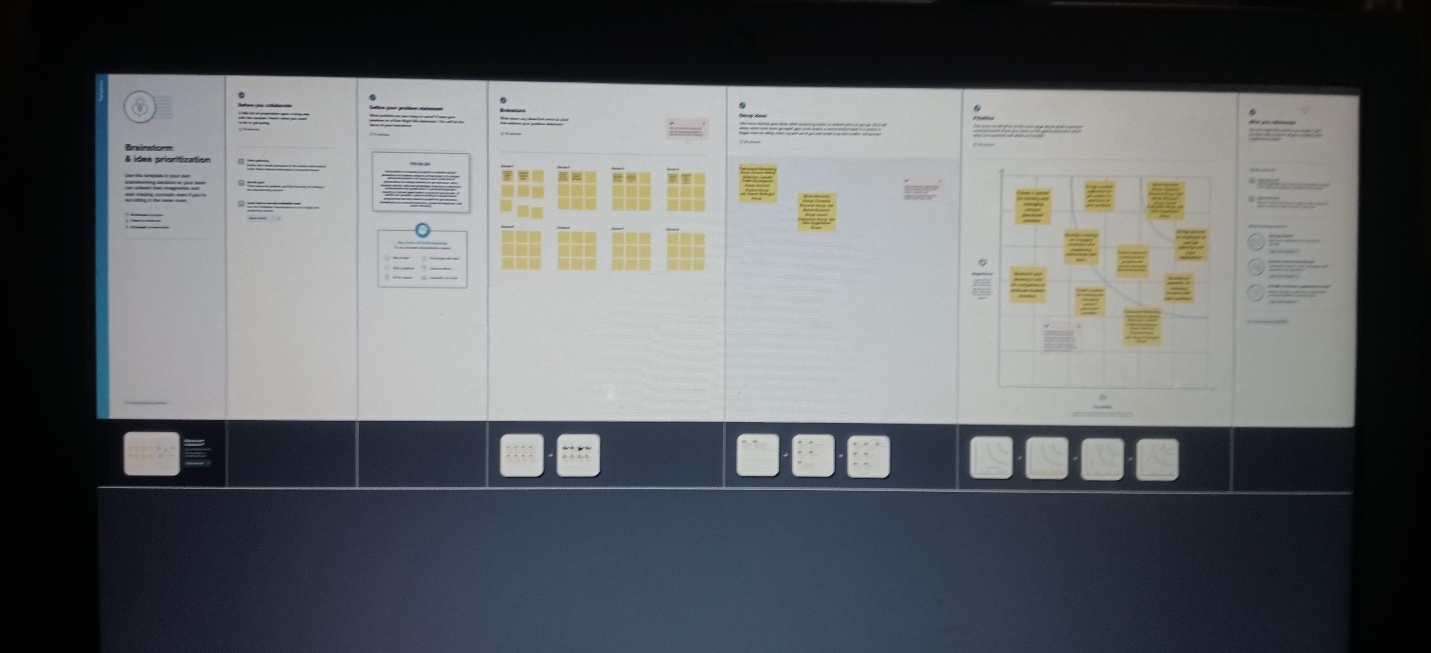
FEAR

CHECK THE WEBSITE



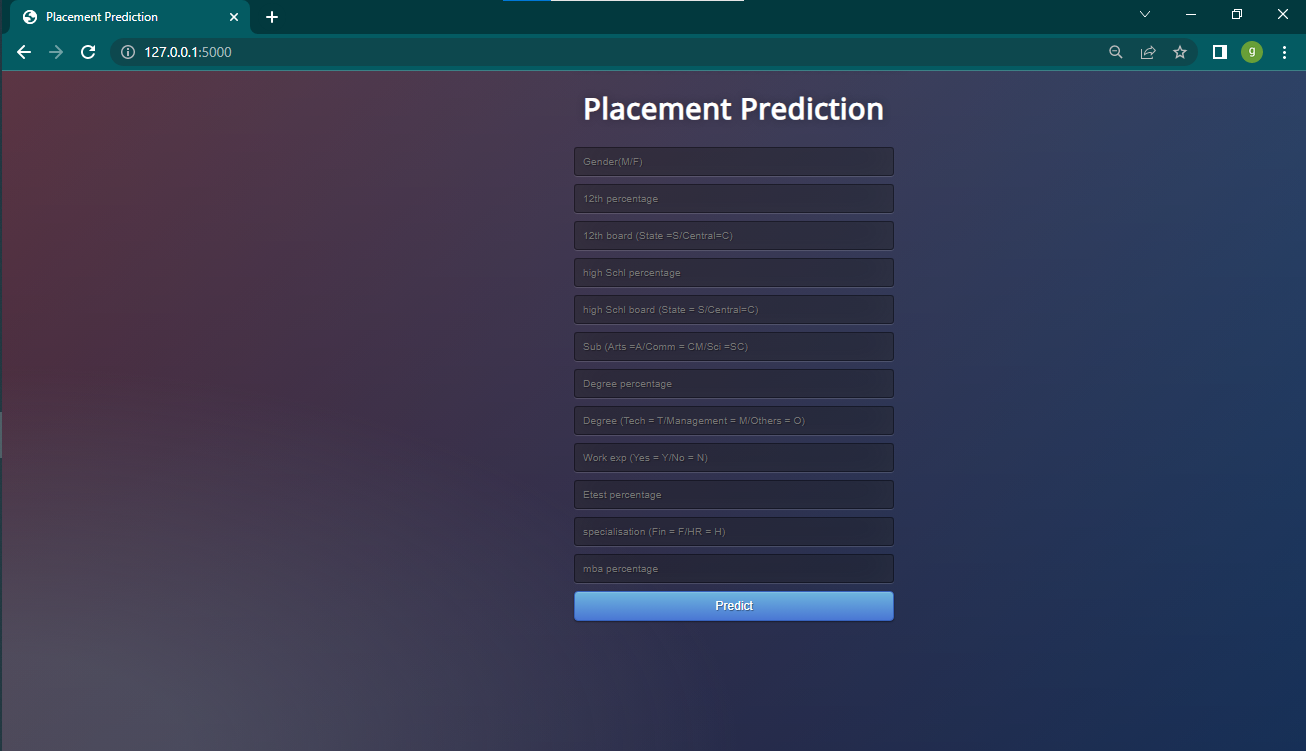
**2.2 IDEATION MAP**



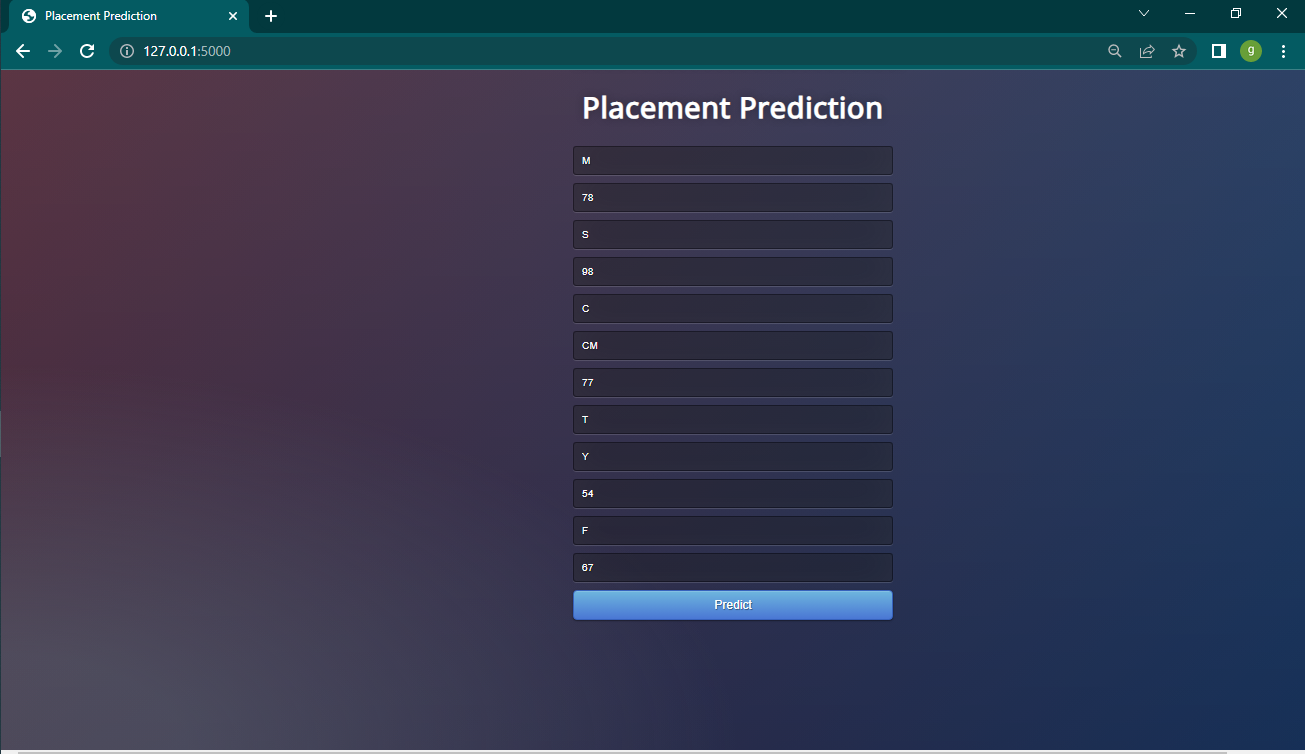
****

**RESULT**

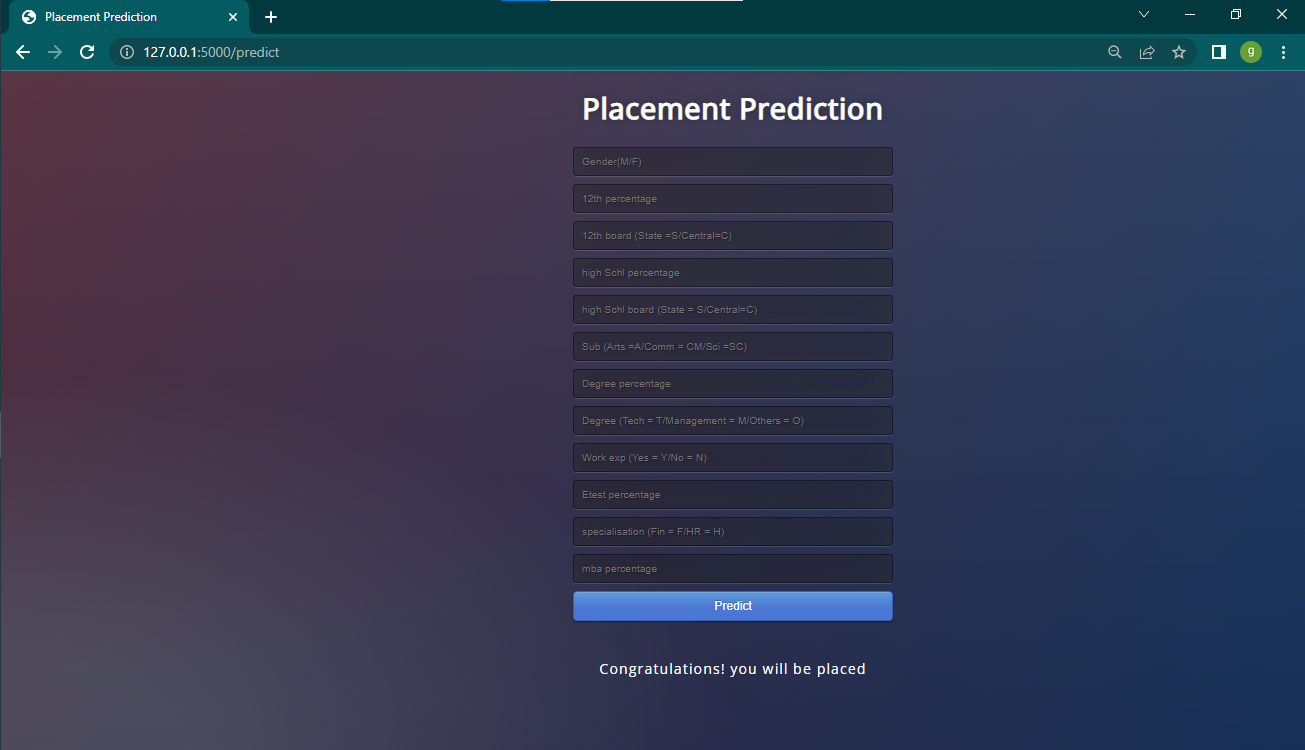
**3.1 RESULT**

****

**HOME PAGE**

****

**INPUT PAGE**

****

**OUTPUT (PREDICTION)**

**ADVANTAGES AND DISADVANTAGES**

**4.1 ADVANTAGES**

* Predict the results and we then compare the efficiency of the algorithms, which is based on the dataset. This model helps the position cell at intervals a corporation to spot the potential students and concentrate to and improve their technical and social skills.
* Increases efficiency of the admission team with zero errors.
* It involves the use of machine learning model of k-nearest neighbor algorithm as base model to classify students or users into appropriate clusters and the result would help them in improving their skills and other mindset.

**4.2 DISADVANTAGES**

* Placement opportunities will not be same for Artificial intelligence and Machine Learning when compared to Computer Science Engineering and Information Technology.
* These Artificial Intelligence and Machine learning are evolving domains it will take some time to get a wide range of exposure.
* The major downside to machine learning is that we are taking personal interaction away from the students. This can dramatically impact their ability to make friends and present themselves well in the workplace over the years ahead.
* Social skills still need to be emphasized even while using machine learning.
* They used normal conditional statements of getting information
* They don’t have proper algorithms for analyzing data for new placements.

**APPLICATIONS**

**5.1 APPLICATIONS**

* Used in Universities.
* Used in high schools.
* Used in all educational institution.
* Used in government institution.
* Used in private institution, therefore it saves more time.

**CONCLUSION**

**6.1 CONCLUSION**

Predicting the placement of a student gives an idea to the Placement Office as well as the student on where they stand. Not all companies look for similar talents. If the strengths and weaknesses of the students are identified it would benefit the student in getting placed. The placement Office can work on identifying the weaknesses of the students and take measures of improvement so that the students can overcome the weakness and perform to the best of their abilities. Thus, the key lies in assessing the capabilities of the student in the right areas and subjecting them to the right training.

**FUTURE SCOPE**

**7.1 FUTURE SCOPE**

Further work can be carried out by applying other algorithms that could lead to improvement in results, also different Skill sets for above mentioned branches and on data of different streams. The key to this would be to identify the aspects of these branches that would test the student’s knowledge in getting placed in a core company for their respective branches.

**APPENDIX**

**8.1 SOURCE CODE**

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pickle

app = Flask(\_\_name\_\_)

model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():

'''

For rendering results on HTML GUI

'''

int\_features = []

for x in request.form.values():

if(x=='M'):

x=1

elif(x=='F'):

x=0

elif(x=='S'):

x=1

elif(x=='C'):

x=0

elif(x=='A'):

x=0

elif(x=='CM'):

x=1

elif(x=='SC'):

x=2

elif(x=='T'):

x=2

elif(x=='M'):

x=0

elif(x=='O'):

x=1

elif(x=='N'):

x=0

elif(x=='Y'):

x=1

elif(x=='H'):

x=1

elif(x=='F'):

x=0

int\_features.append(int(x))

final\_features = [np.array(int\_features)]

prediction = model.predict(final\_features)

output = round(prediction[0], 2)

if(output==1):

return render\_template('index.html', prediction\_text='Congratulations! you will be placed ')

if(output==0):

return render\_template('index.html', prediction\_text='keep up the hardwork! Placement chances are low ')

@app.route('/predict\_api',methods=['POST'])

def predict\_api():

'''

For direct API calls trought request

'''

data = request.get\_json(force=True)

prediction = model.predict([np.array(list(data.values()))])

output = prediction[0]

return jsonify(output)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)