VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

IBM PROJECT REPORT

PROJECT TITLE :University Admit Eligibility Predictor

TEAM ID : PNT2022TMID22967

TEAM MEMBERS : SANJAY G M

ASHIKA M

ELANGO K

JANANI A

Github link : https://github.com/IBM-EPBL/IBM-Project-36150-1660293210

INTRODUCTION

Overview :

The project is implemented using applied data science model that predicts whether the user is eligible for an admission in the selected rated universities with provided details such as marks and others. Majority of universities follow similar guidelines for providing admission to students. Universities take into consideration different factors like score on aptitude based examination like the General Record Examination (GRE), command over the English language is judged based on their score in English competency test like Test Of English as a Foreign Language (TOEFL) OR International English Language Testing System (IELTS), their work experience in same or other fields, the quality of the Letters Of Recommendation (LOR) and the Statement Of Purpose documents provided by the student etc. Based on the overall profile of the student decision is taken by the universities admission team to admit or reject a particular candidate.The algorithm works in such a way that when the user provides the details such as (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research) the percentage of chance of admit is displayed. The user is provided with a UI (Web based application) in which the user can enter the details mentioned above for prediction. The main advantage of this is that the user can avoid long process of having to check the eligibility of a university admission by himself and make use of this application to predict the eligibility / chance of admit.

Purpose :

The purpose of this project is to make the prediction of eligibility of an admission to a rated university with ease using a UI with the provided user details (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research). The task of shortlisting the universities where the student has high chances of admission is difficult for the students, so they end up with applying to many universities in hopes of getting admission in few of them thus investing an extra amount of money in the applications. Most of the students dont take the risk of evaluating the colleges by themselves, and they seek the help of the education consultancy firms to do it for them. Again for this students have to pay a huge amount of fee to the education consultant.This also eliminates the possibility of human errors. Also, the system will provide a recommendation of universities to the student to which the student has a high possibility of getting admission.

College admission predictor is a boon to many students. This helps the student not only to help in filling out the application forms but also give the students an idea about their future college by calculating their cut off.

- When students come from rural places , they find it hard to go along with the formal procedures. So, this application helps them a lot and eases out their fear.
- Whatever may be their scores , this application helps to find the best colleges . Hence, our proposed computer aided system will help the students to get the list of all colleges in which they could get the admission at the click of a button.
- The students only have to enter their marks of XII, AIEEE etc. With this

application, the students can very easily obtain the list of colleges even branch wise and course wise. This will not only make the admission process easy but also minimizes stress for students . The main objective of our system is to make the right choice of colleges.

LITERATURE SURVEY

Existing problem :

GRADE system was developed by (Waters and Miikkulainen (2013)) to support the admission process for the graduate students in the University of Texas Austin Department of Computer Science. The main objective of the project was to develop a system that can help the admission committee of the university to take better and faster decisions. Logistic regression and SVM were used to create the model, both models performed equally well and the final system was developed using Logistic regression due to its simplicity. The time required by the admission committee to review the applications was reduced by 74% but human intervention was required to make the final decision on status if the application.(Nandeshwar et al. (2014)) created a similar model to predict the enrolment of the student in the university based on the factors like SAT score, GPA score, residency race etc. The Model was created using the Multiple Logistic regression algorithm, it was able to achieve accuracy rate of 67% only.

Today in college's student details are entered manually .The student details in separate records are tedious task .Referring to all these records updating is needed .There is a

chance for more manual errors.

1. When the student comes in college.
2. First of all,he/she takes admission form from reception.
3. Fills it and submits it into office.
4. Filled form is first checked with documents like merit list an details came from universitY and verified by an official person ,if there is any mistake then it is corrected.
5.Atthetimeofsubmissionofitthefeesisdepositedbythecandidate.
6. At the time of submission of admission form admission number is assignedtothecandidate by the institute.
7. Candidate gets the receipt of fees deposition.

DISADVANTAGES OF THIS SYSTEM :

1. Require much man power i.e. much efforts , much cost and hard to operate and maintain.
2. Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

Also, Previous research done in this area used Naive Bayes algorithm which will evaluate the success probability of student application into a respective university but the main drawback is they didn't consider all the factors which will contribute in the student admission process like TOEFL/IELTS, SOP, LOR and under graduate score. Nave Bayes algorithm was used to evaluated based on their accuracy to select the best candidates for the college. Limitation of this research as that it did only relied on the GRE, TOEFL and Undergraduate Score of the student and missed on taking into

consideration other important factors like SOP and LOR documents quality, past work experience, technical papers of the students etc. ze progress of prospective students by comparing the score of students currently studying at university. The model thus predicted whether the aspiring student should be admitted to university on the basis of various scores of students. Since the comparisons are made only with students who got admission into the universities but not with students who got their admission rejected so this method will not be that much accurate.

Proposed solution :

These problems can be resolved by using regression algorithms / classification algorithms as they can consider most of the features for prediction. Linear regression / KNN classification / Random forest Regressor can be used as the machine learning model for the model. XG boost model can also be used which performs better on small to medium scale datasets but the model giving accurate and desired results only will be selected.

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools.

Following are the objectives of the proposed system:

- **Reach to geographically scattered student.**
- **Reducing time in activities.**
- **Paperless admission with reduced man power.**

- **Operational efficiency.**
- **Manage large number of student details.**
- **Manage all details of student who registered for the course.**
- **Activities like updating, modification, deletion of records should be easier.**
- **Reducing time in activities:** Reduce the time taken process the applications of students admitting a student, conducting the online examination, verify student marks, and send call letters to selected students.
- **Centralized data handling** :Transfer the data smoothly to all the departments involved and handle the data centralized way.
- **Paperless admission with reduced man power** : Reduce the manpower needed to perform all the admission and administration task by reducing the paper works Cost cutting. Reduce the cost involved in the admission process.
- **Operational efficiency**: Improve the operational efficiency by improving the quality of the process.

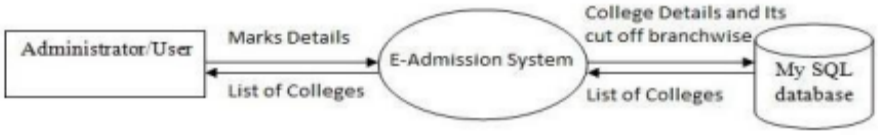


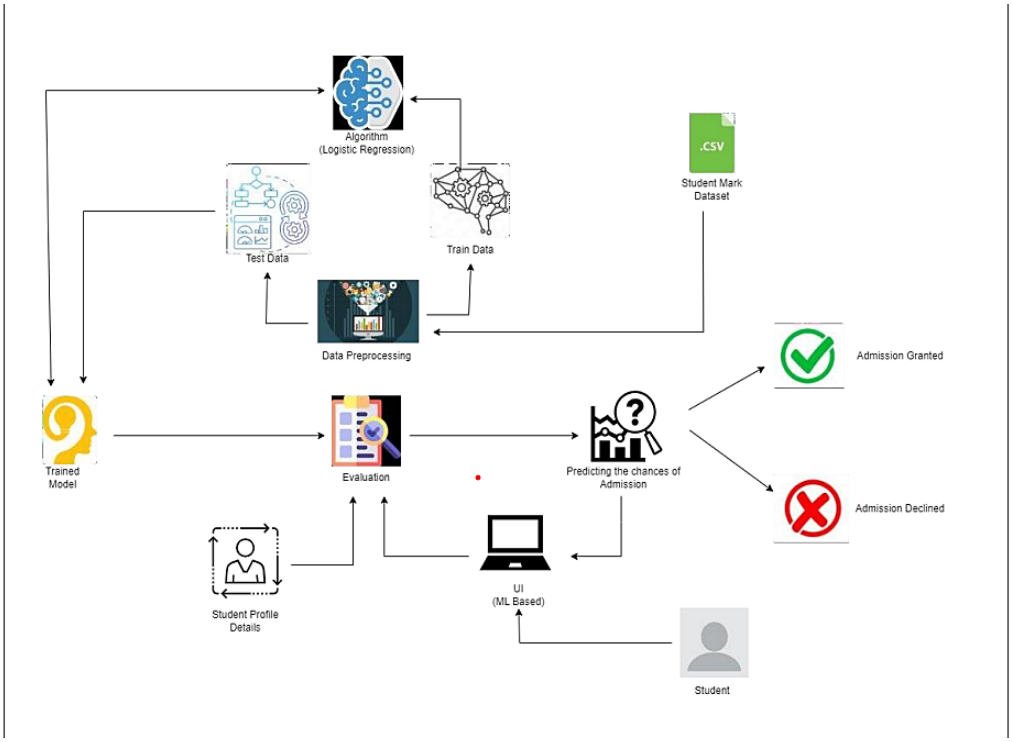
Fig:1 Dataflow Diagram

ADVANTAGES OF PROPOSED SYTSEM:

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous

THEORETICAL ANALYSIS

Block Diagram :



Hardware/Software designing :

ITEM	CONTENT
CPU	Intel i3 7th gen or above/ AMD Ryzen 3500 or above
GPU	Integrated / Dedicated (atleast 2GB)
RAM	8 GB
Operating system	Windows 10/ Linux / MacOS
Programming language	Python 3.9
ML library	Scikit learn
Other libraries	Matplot,seaborn,numpy,pandas,pickle

Research Methodology :

Business Understanding : Initially good amount of time was spent on understanding the problem statement by understanding the concerns of students regarding the current application process, the objectives of the research were defined in this process.

Data Understanding : Data required for the research was collected from multiple data sources. Different features of the data were analyzed based on their importance and relevance. Data-set would be explained in more detail further.

Data Preparation : In this phase, the data from multiple data sources were integrated into a final data-set. Further the data was cleaned by removing unwanted columns, performing transformation and cleaning activities on the data.

Modelling : Multiple machine learning models were developed to predict the likelihood of success of the student's application in a particular university. The user interface was developed to allow the users to access these models.

Evaluation : Models developed were evaluated based on their performance and accuracy. More information will be presented in the evaluation section of the paper.

Deployment : Once the models were evaluated they were integrated with code developed for user interface using the Shiny package in R.

TESTING OBJECTIVES :

There are several rules that can serve as testing objectives. They are: Testing is process of executing a program and finding a bug. A good test case is one that has a high probability of finding an undiscovered

A successful test is one that uncovers an undiscovered error .If testing is conducted

successfully according to the objectives as stated above ,it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to havebeenmet.

UNIT TESTING :

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors

SYSTEM TESTING :

In this, the entire system was tested as a whole with all forms, code, modules and class modules .System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions. System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

INTEGRATION TESTING :

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed .Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines ,after linking, may not do the desired function

expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time ,conducting test to uncover errors associated with the interface. In the testing the programs are constructed and tested in the small segments.

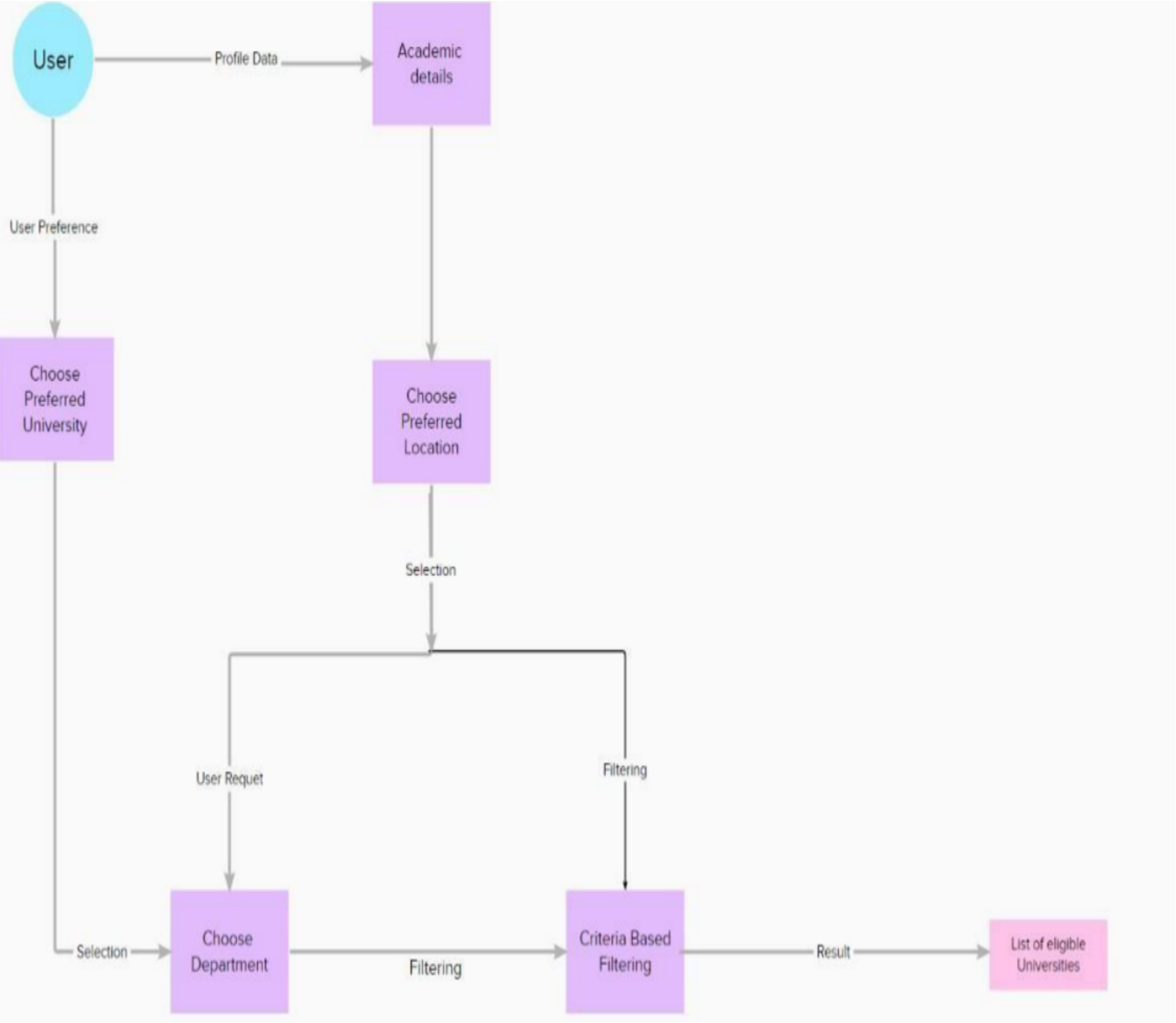
WHITE BOX TESTING :

White-box testing is concerned with testing the implementation of the program .The intentofthistestingisnottoexercise allthedifferent input oroutput conditions butto exercise the different programming structures and data structures used in the program. White box testing is also called structural testing .To test the structure of a program, structural testing aims to achieve test cases that will force the desired coverage of different structures. Various criteria have been proposed for this there are three different approaches to structural testing: control flow-based testing data flow-based testing, and mutation testing.

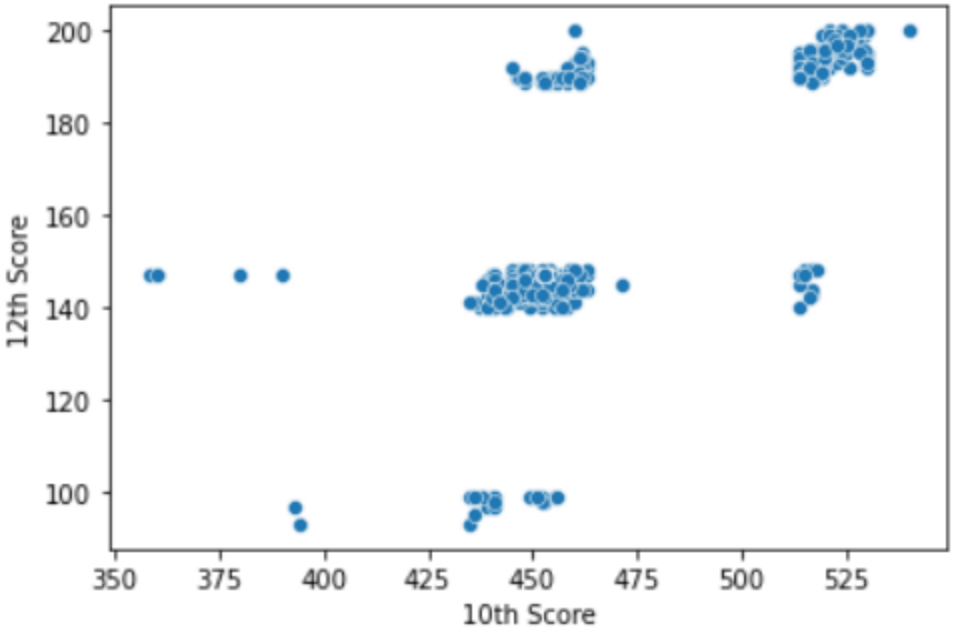
BLACK BOX TESTING :

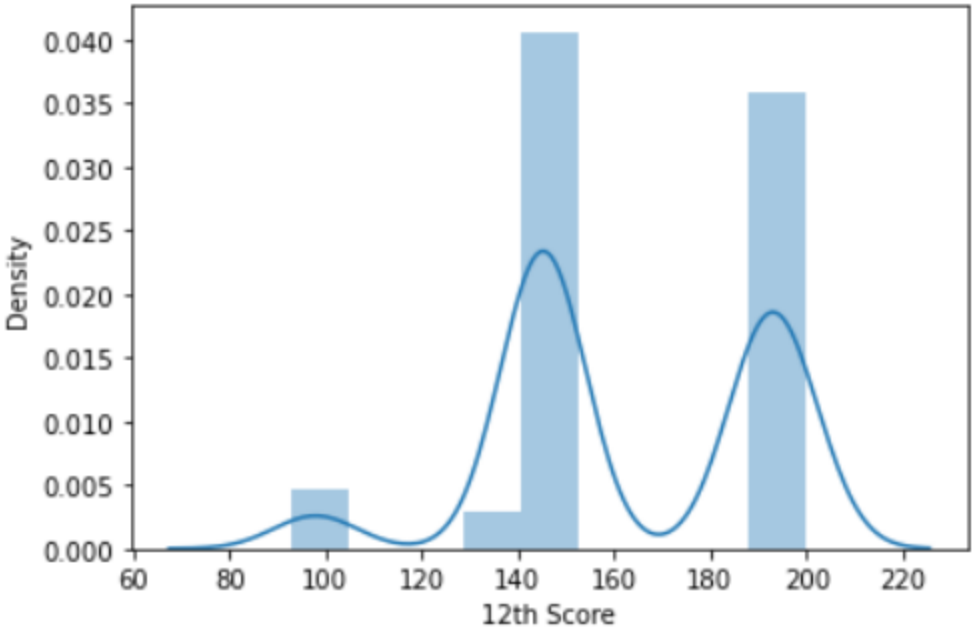
In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module ,and the internals of the module or the program are not considered for selection of test cases .In black-box testing, the tester only knows the inputsthatcanbegiventothe systemandwhatoutputthe systemshould give.Thisformoftesting is also called functional or behavioral testing .The most obvious functional testing procedure is exhaustive testing. One criterion for generating test cases is to generate them randomly. There are no formal rules for designing test cases for functional testing.

Data Flow Diagram

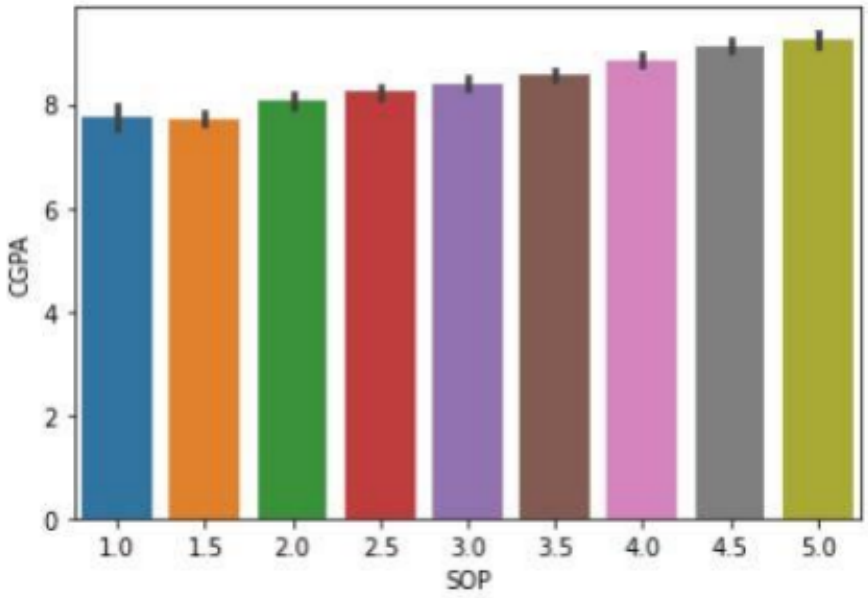


EXPERIMENTAL ANALYSIS

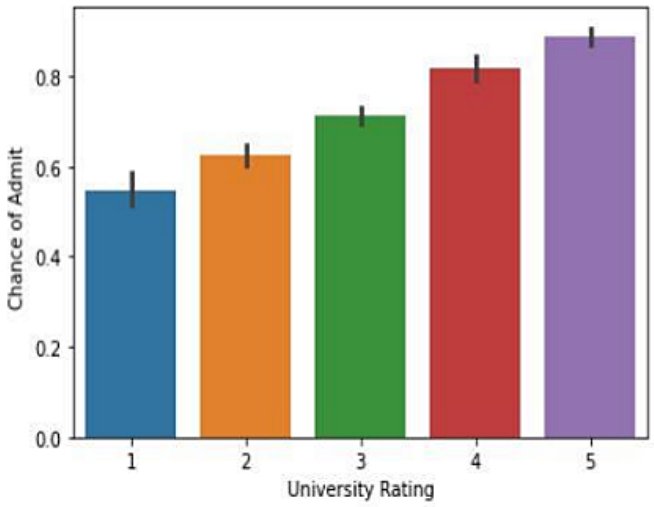




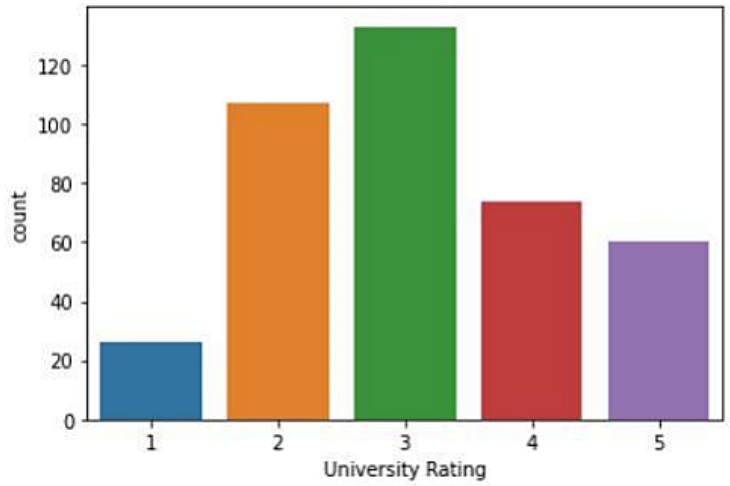
```
<AxesSubplot:xlabel='SOP', ylabel='CGPA'>
```



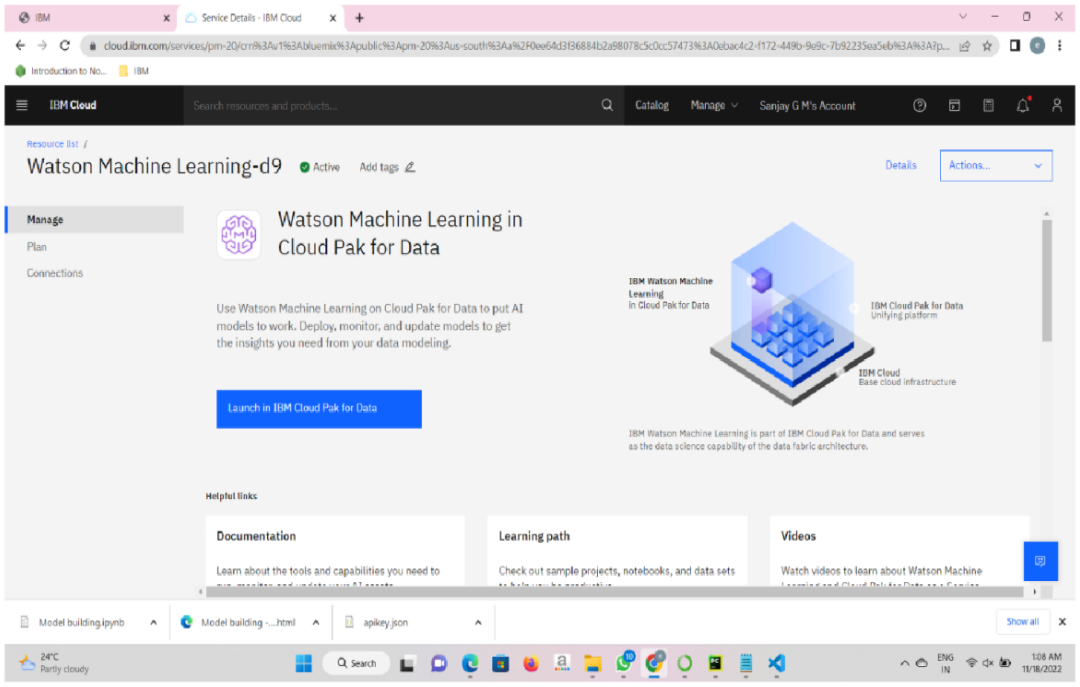

```
<AxesSubplot:xlabel='University Rating', ylabel='Chance of Admit '>
```



```
<AxesSubplot:xlabel='University Rating', ylabel='count'>
```

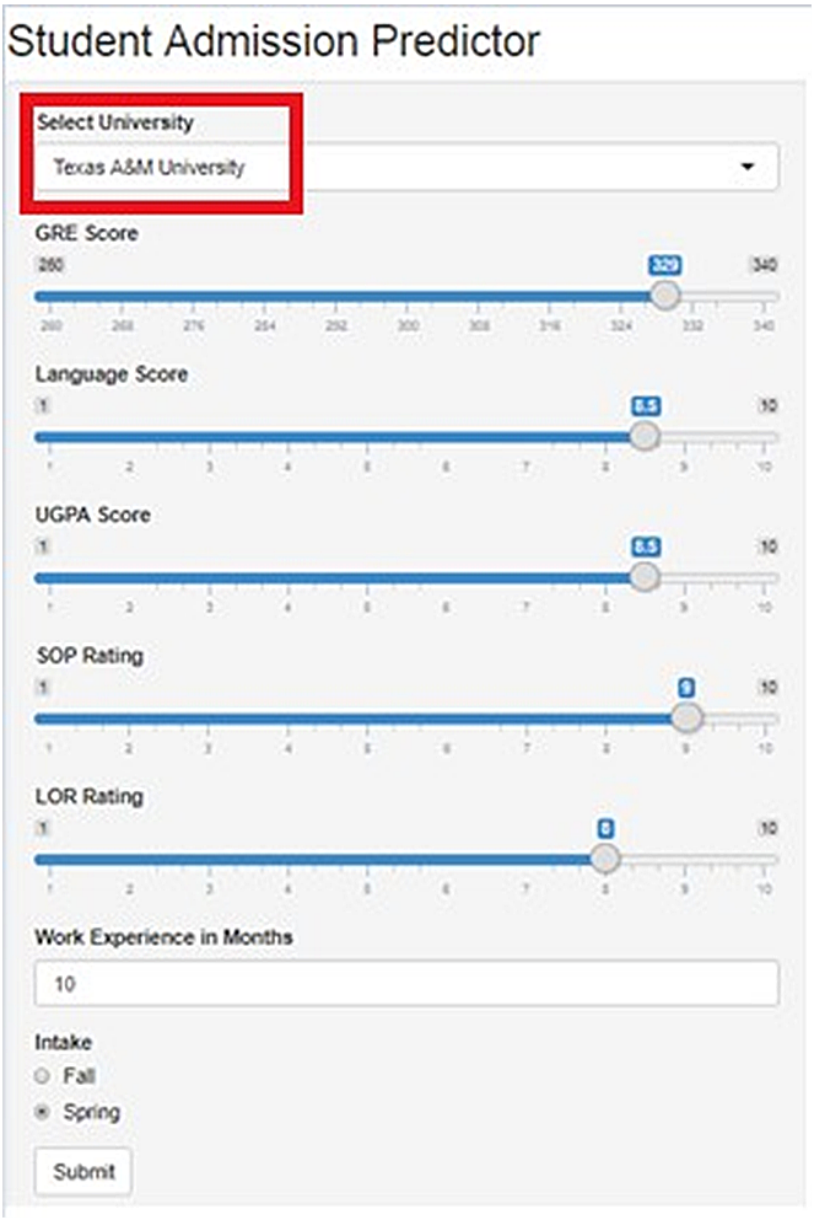


CLOUD STORAGE :



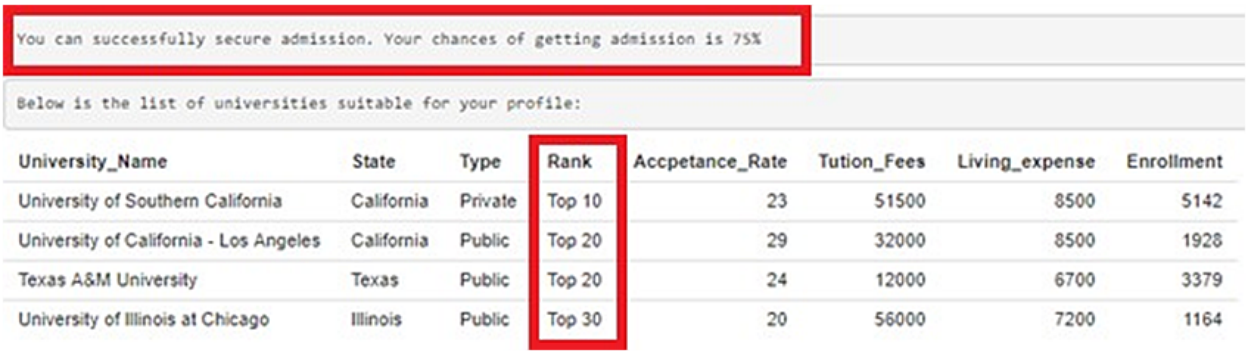
Case Study :

When a student with top grades evaluates his/her profile for admission in colleges Ranked in Top 10 to Top 40 in the USA. Below is the screen-shot of the user interface before submitting the student profile.



Below is the output, showing the student has 75% chance of getting admission in the Texas A&M University. It also suggests the student to try

applying for the other universities listed. As the gradesof the students are very good, universities ranked top 10 to top 40 were recommended. Also other important details regarding the recommended universities like State,Type, Rank, Acceptance Rate, Tuition Fees, Living Expense and Yearly Enrolmentare provided to the student.



You can successfully secure admission. Your chances of getting admission is 75%

Below is the list of universities suitable for your profile:

University_Name	State	Type	Rank	Accpetance_Rate	Tution_Fees	Living_expense	Enrollment
University of Southern California	California	Private	Top 10	23	51500	8500	5142
University of California - Los Angeles	California	Public	Top 20	29	32000	8500	1928
Texas A&M University	Texas	Public	Top 20	24	12000	6700	3379
University of Illinois at Chicago	Illinois	Public	Top 30	20	56000	7200	1164

fig : case study output

CONCLUSION :

This system, being the first we have created in PHP, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methodsto improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be.Since so many parts of the admission system are integrated in to

one another, if an error occurs on one page, it may bead is play error,for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first .Language used must be simple and easy to understand and compatibility is paramount.If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability .Overall, the system performs well, and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

BIBLIOGRAPHY

- L. Chang , Applying Data Mining to Predict College Admissions Yield, Chapter 4 in J. Luan and C. Zhao (Eds.), Data mining in action: Case studies, Spring 2008 - College of Education.
- Rensong Dong,The module of prediction of College Entrance Examination aspiration, Fuzzy Systems and Knowledge Discovery (FSKD), 31 May 2012 ,1559-1562.
- Borah M.D., Application of knowledge based decision technique to predict student enrolment decision , Recent Trends in Information Systems (Re TIS), 21-23 Dec. 2011,180-184.

APPENDIX

Source code :

```
    <HEAD>

        <STYLE>
            @import url('https://fonts.googleapis.com/css?family=Open+Sans&display=swap');


/*glow for university admission prediction dont change*/

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .prev, .next,.text {font-size: 11px}
}

.glow {
  font-size: 80px;
  color: #fff;
  text-align: center;
  animation: glow 1s ease-in-out infinite alternate;
}

@keyframes glow {
  from {
    text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #e60073, 0 0 40px #e60073, 0 0 50px #e60073
  }

  to {
    text-shadow: 0 0 20px #fff, 0 0 30px #ff4da6, 0 0 40px #ff4da6, 0 0 50px #ff4da6, 0 0 60px #ff4
     }
}


.glow-on-hover {
    width: 220px;
```

```
    height: 50px;
    border: none;
    outline: none;
    color: #fff;
    background: rgb(161, 15, 117);
    cursor: pointer;
    position: relative;
    z-index: 0;
    border-radius: 10px;
}

.glow-on-hover:before {
    content: '';
    background: linear-gradient(45deg, #ff0000, #ff7300, #fffb00, #48ff00, #00ffd5, #002bff, #7a00f
    #ff0000);
    position: absolute;
    top: -2px;
    left:-2px;
    background-size: 400%;
    z-index: -1;
    filter: blur(5px);
    width: calc(100% + 4px);
    height: calc(100% + 4px);
    animation: glowing 20s linear infinite;
    opacity: 0;
    transition: opacity .3s ease-in-out;
    border-radius: 10px;
}

.glow-on-hover:active {
    color: rgb(108, 14, 107)
}

.glow-on-hover:active:after {
    background: transparent;
}

.glow-on-hover:hover:before {
    opacity: 1;
}

.glow-on-hover:after {
    z-index: -1;
```

```
    content: '';
    position: absolute;
    width: 100%;
    height: 100%;
    background: #e60073;
    left: 0;
    top: 0;
    border-radius: 10px;
}

@keyframes glowing {
    0% { background-position: 0 0; }
    50% { background-position: 400% 0; }
    100% { background-position: 0 0; }
}

/* university admission prediction don't change*/




:root{
    --succes-color: #2ecc71;;
    --error-color: #e74c3c;
}


*{
    box-sizing: border-box;
}

body{

    background-color: #573251;
    font-family: 'Open Sans', sans-serif;
    display: flex;
    align-items: center;
    justify-content: center;
    min-height: 100vh;
    margin: 0;
}
```

```
.container{
    background-color: #573251;
    border-radius: 5px;
    box-shadow: 0 2px 10px rgba(216, 90, 205, 0.3);
    width: 1000px;
}

h2{
    text-align: center;
    margin: 0 0 20px;
}

.form{
    padding: 30px 40px;
}

.form-control{
    margin-bottom: 10px;
    padding-bottom: 20px;
    position: relative;
}

.form-control label{
    color:rgb(255, 255, 255);
    display: block;
    margin-bottom: 5px;
}

.form-control input
{
    border: 1px solid #f0f0f0;
    border-radius: 4px;
    display: block;
    width: 100%;
    padding: 10px;
    font-size: 10px;
}

.form-control input:focus{
    outline: 0;
    border-color: #777;

}
```

```
.form-control.success input {
    border-color: var(--succes-color);
}

.form-control.error input {
    border-color: var(--error-color);
}

.form-control small{
    color: var(--error-color);
    position: absolute;
    bottom: 0;
    left: 0;
    visibility: hidden;
}

.form-control.error small{
    visibility: visible;
}
.form button {
    cursor: pointer;
    background-color: #3498db;
    border: 2px solid #3498db;
    border-radius: 4px;
    color: #fff;
    display: block;
    padding: 10px;
    font-size: 16px;
    margin-top:20px;
    width:100%;
}
        </STYLE>
        </HEAD>
        <BODY>

<div class="container">

        <form id="form" class="form">
            <h1 class="glow" style="font-size:50px">UNIVERSITY ADMISSION PREDICATOR</h1>
            <div class="form-control">
                <label for="username">Username</label>
                <input type="text" id="username" placeholder="Enter Username">
```

```
    <small>Error Message</small>
</div>
<div class="form-control">
    <label for="email">Email</label>
    <input type="text" id="email" placeholder="Enter email">
    <small>Error Message</small>
</div>
<div class="form-control">
    <label for="email">Date Of Birth</label>
    <select name="Birthday_day" id="Birthday_Day">
        <option value="-1">Day:</option>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>

        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>

        <option value="13">13</option>
        <option value="14">14</option>
        <option value="15">15</option>
        <option value="16">16</option>
        <option value="17">17</option>
        <option value="18">18</option>
        <option value="19">19</option>
        <option value="20">20</option>
        <option value="21">21</option>

        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
        <option value="25">25</option>
        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
```

```
    <option value="30">30</option>

    <option value="31">31</option>
    </select>

<select id="Birthday_Month" name="Birthday_Month">
    <option value="-1">Month:</option>
    <option value="January">Jan</option>
    <option value="February">Feb</option>
    <option value="March">Mar</option>
    <option value="April">Apr</option>
    <option value="May">May</option>
    <option value="June">Jun</option>
    <option value="July">Jul</option>
    <option value="August">Aug</option>
    <option value="September">Sep</option>
    <option value="October">Oct</option>
    <option value="November">Nov</option>
    <option value="December">Dec</option>
    </select>

    <select name="Birthday_Year" id="Birthday_Year">

    <option value="-1">Year:</option>
    <option value="2012">2012</option>
    <option value="2011">2011</option>
    <option value="2010">2010</option>
    <option value="2009">2009</option>
    <option value="2008">2008</option>
    <option value="2007">2007</option>
    <option value="2006">2006</option>
    <option value="2005">2005</option>
    <option value="2004">2004</option>
    <option value="2003">2003</option>
    <option value="2002">2002</option>
    <option value="2001">2001</option>
    <option value="2000">2000</option>

    <option value="1999">1999</option>
    <option value="1998">1998</option>
    <option value="1997">1997</option>
    <option value="1996">1996</option>
    <option value="1995">1995</option>
```

```
        <option value="1994">1994</option>
        <option value="1993">1993</option>
        <option value="1992">1992</option>
        <option value="1991">1991</option>
        <option value="1990">1990</option>

        <option value="1989">1989</option>
        <option value="1988">1988</option>
        <option value="1987">1987</option>
        <option value="1986">1986</option>
        <option value="1985">1985</option>
        <option value="1984">1984</option>
        <option value="1983">1983</option>
        <option value="1982">1982</option>
        <option value="1981">1981</option>
        <option value="1980">1980</option>
        </select>

    <small>Error Message</small>
</div>
<div class="form-control">
    <label for="username">Mobile Number</label>
    <input type="text"placeholder="Enter Mobile Number" maxlength="10">
    <small>Error Message</small>
</div>
<div class="">
    <label style="color:rgb(255, 255, 255)">GENDER</label>
    <input type="radio" name="Gender" value="Male" /><span style="color:#777">MALE</spa
    <input type="radio" name="Gender" value="Male" /><span style="color:#777">Female </
</div><br>
<div class="form-control">
    <label style="color:rgb(255, 255, 255)">ADDRESS</label>
    <textarea name="Address" placeholder="Enter Full Address for srutaning" rows="4" co
</div>
<div class="form-control">
    <label >City</label>
    <input type="text" placeholder="City" maxlength="10">
    <small>Error Message</small>
</div>
<div class="form-control">
    <label >Pincode</label>
    <input type="text"placeholder="Pincode" maxlength="6">
    <small>Error Message</small>
```

```
            </div>
            <div class="form-control">
                <label >State</label>
                <input type="text"placeholder="State" maxlength="6">
                <small>Error Message</small>
            </div>
            <div class="form-control">
                <label >Country</label>
                <select >
                    <option for="city">Indian</option>
                    <option for="city">Forigner</option>
                  </select>
                <small>Error Message</small>
            </div>
        </form>
<head>
    <style>

        </style>
    </head>
            <form action="target.html" method="POST">
            <div class="form-control">

                <table>


                    <tr>
                    <td align="center"><b style="color:rgb(255, 255, 255)">Sl.No.</b></td>
                    <td align="center"><b style="color:rgb(255, 255, 255)">Examination</b></td>
                    <td align="center"><b style="color:rgb(255, 255, 255)">Board</b></td>
                    <td align="center"><b style="color:rgb(255, 255, 255)">Percentage</b></td>
                    <td align="center"><b style="color:rgb(255, 255, 255)">Year of Passing</b></td>
                    </tr>

                    <tr>
                    <td style="color:rgb(162, 159, 159)">1</td>
                    <td style="color:rgb(162, 159, 159)">Class X</td>
                    <td><input type="text" name="ClassX_Board" maxlength="30" size="40" /></td>
                   <td> <input type="number" id="quantity" name="10" placeholder="Percentage/500" m
                    <td><input type="text" name="ClassX_YrOfPassing" maxlength="30" size="40" /></t
                    <small>Error Mesage</small>
                    </tr>
```

```
                    <tr>
                    <td style="color:rgb(162, 159, 159)">2</td>
                    <td style="color:rgb(162, 159, 159)">Class XII</td>
                    <td><input type="text" name="ClassXII_Board" maxlength="30"  size="40"/></td>
                     <td><input type="number" name="12" placeholder="cutoff/200"min="0" max="200" s
                    <td><input type="text" name="ClassXII_YrOfPassing" maxlength="30" size="40" /><
                    </tr>

                    <tr>
                    <td style="color:rgb(162, 159, 159)">3</td>
                    <td style="color:rgb(162, 159, 159)">Diploma Graduation</td>
                    <td><input type="text" name="Graduation_Board" maxlength="30" size="40"/></td>
                    <td><input type="number" id="hi_extra" name="col" placeholder="Percentage/100"
                    <td><input type="text" name="Graduation_YrOfPassing" maxlength="30" size="40"/>
                    </tr>

                    <tr>
                    <td style="color:rgb(162, 159, 159)">4</td>
                    <td style="color:rgb(162, 159, 159)">Under Graduation</td>
                    <td><input type="text" name="Masters_Board" maxlength="30"size="40" /></td>
                    <td><input type="number" name="ClassX_Board" min="0" max="100" Placeholder="CGP
                    <td><input type="text" name="Masters_YrOfPassing" maxlength="30"size="40" /></t
                    </tr>
                    <tr>
                        <td> </td><td style="color:rgb(255, 255, 255)"><b>Extracurricular</b></td><
255)"><b>Rating</b></td>
                    </tr>
                    <tr>
                        <td style="color:rgb(162, 159, 159)">5</td>
                    <td style="color:rgb(162, 159, 159)">Sports</td>
                    <td><input type="text" name="Masters_Board" maxlength="30"size="40" /></td>
                    <td><input type="number" name="se" placeholder="1 - 5" style="width:305px; heig
                    </tr>
                    <tr>
                        <td style="color:rgb(162, 159, 159)">6</td>
                    <td style="color:rgb(162, 159, 159)">Clubs</td>
                    <td><input type="text" name="Masters_Board" maxlength="30"size="40" /></td>
                    <td><input type="number" name="c" placeholder="1 - 5" style="width:305px; heigh
                    </tr>
                    <tr>
                        <td style="color:rgb(162, 159, 159)">7</td>
                    <td style="color:rgb(162, 159, 159)">Research</td>
                    <td><input type="text" name="Masters_Board" maxlength="30"size="40" /></td>
```

```
                    <td><input type="number" name="r" placeholder="0 - 1" style="width:305px; heigh
                    </tr>
                    </table>


            <center>

                <div style="color:#e60073">
                  <input type="submit"  class="glow-on-hover"  onclick="demo()" value="Check Your L
                </div>
                </center>

            </div>
        </form>
</div>
        <SCRIPT>
            function demo(){
                var i = document.getElementById("hi_extra").value;
                i = i/10;
                i = i/2;
                i = Math.round(i);
                document.getElementById("hi_extra").value=i;
            }
            const form = document.getElementById('form');
const username = document.getElementById('username');
const email = document.getElementById('email');


//Show input error messages
function showError(input, message) {
    const formControl = input.parentElement;
    formControl.className = 'form-control error';
    const small = formControl.querySelector('small');
    small.innerText = message;
}

//show success colour
function showSucces(input) {
    const formControl = input.parentElement;
    formControl.className = 'form-control success';
}

//check email is valid
```

```
function checkEmail(input) {
    const re = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]
    if(re.test(input.value.trim())) {
        showSucces(input)
    }else {
        showError(input,'Email is not invalid');
    }
}


//checkRequired fields
function checkRequired(inputArr) {
    inputArr.forEach(function(input){
        if(input.value.trim() === ''){
            showError(input,`${getFieldName(input)} is required`)
        }else {
            showSucces(input);
        }
    });
}


//check input Length
function checkLength(input, min ,max) {
    if(input.value.length < min) {
        showError(input, `${getFieldName(input)} must be at least ${min} characters`);
    }else if(input.value.length > max) {
        showError(input, `${getFieldName(input)} must be les than ${max} characters`);
    }else {
        showSucces(input);
    }
}

//get FieldName
function getFieldName(input) {
    return input.id.charAt(0).toUpperCase() + input.id.slice(1);
}



//Event Listeners
form.addEventListener('submit',function(e) {
    e.preventDefault();
```

```
    checkRequired([username, email, password, password2]);
    checkLength(username,3,15);
    checkLength(password,6,25);
    checkEmail(email);
    checkPasswordMatch(password, password2);
});

            </SCRIPT>


    </BODY>
    </HTML>
```