A Database Mini-Project Report

on

“Automated Attainment Generation”

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

by

33249 Harsh Sakhrani

33250 Saloni Parekh

33254 Muskaan Shraogi

33259 Tanmay Pardeshi

Department Of Information Technology

Pune Institute of Computer Technology College of Engineering

Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

A.Y. 2020-2021

ACKNOWLEDGEMENT

We thank everyone who has helped and provided valuable suggestions for successfully creating a wonderful project.

We are very grateful to Mrs. Sumitra Jakhete who is our guide for the project, and our principal Dr. P. T. Kulkarni. They have been very supportive and have ensured that all facilities remained available for smooth progress of the project.

We would like to thank our professor Dr. Emmanuel M. for providing us the theoretical knowledge for the project. We would also like the entire DBMS staff team for providing valuable reviews and suggestions from time to time.

We would like to thank our entire department and college staff for the very valuable help and coordination throughout the duration of the project.

We would also like to thank our families and all our friends for the valuable support they provided throughout the duration of the project.

Harsh Sakhrani

Muskaan Shraogi

Saloni Parekh

Tanmay Pardeshi

CONTENTS

1.ABSTRACT

Calculation of termwork, internal scores and attainment is one of the more essential parts of the working of any institute. Trying to automate this process is what our project focuses on. We have created an extremely intuitive portal which makes it easier for the staff members to communicate with their respective subject heads and keep a track of individual student assessment. We used MySQL as the database backend along with NodeJS and ExpressJS. React was used to make a clean interface.

2.INTRODUCTION:

This report discusses the method and result of the work done in the development of Target - A platform developed to automate the process of target attainment calculation and provide easy storage for the same.
The major building blocks used for developing Target are as follows:

- MySQL
- ExpressJS
- ReactJS
- NodeJS

3.BACKGROUND AND MOTIVATION:

Attainment calculation is a time consuming, tedious and logically intensive task. This is especially true when it has to be done time and again and therefore has immense scope for automation. The automation would bring about almost the same results and would be an easy and a convenient way to assess students repeatedly without any errors, both of which cannot be achieved manually at least in the same amount of time.

The fact that this project could be made use of in the practical scenario of our IT Department in PICT was the motivation behind us choosing this problem statement. It would serve the dual purpose of us gaining more knowledge about DBMS as well as the creation of something useful for our college.

4.OBJECTIVE:

- To make the calculation of attainment effortless and quick
- To make the calculation seamless and accurate
- To keep record of the attainment calculated
- To manage and organise the allotment of divisions and subjects to teachers

5.METHODOLOGY:

- **Overview:**
  The platform requires registration of staff members so as to create a portal for the administrator to allocate subjects and co-ordinator roles to them.
  The teachers are responsible for the submission of the worksheet with marks before the given deadline as per the given format so as to continue with the further process.
  The subject coordinators have the privilege to view an interface where they can see details of their subjects and get to send emails to teachers who have not submitted the report within the deadline. They also have the access to view the calculated results provided all teachers have submitted their worksheet and download it in the form of a report.

- **Calculation of Attainment:**
  The target values for mid-term attainment (MT) and end-term attainment (SPPU) are provided by the subject coordinator. These values are used as parameters to determine the CO attainment for the current academic year by incorporating them into a set of formulae for each division. The target attainment for individual subjects is calculated by aggregating the attainment calculated for each division.
  The following is the method used for calculating the attainment:

A. Attainment through university exam:

UA_AT_L1	**90% students scoring more than 40% in a specified course in SPPU exam.**
UA_AT_L2	40% students scoring more than 60% in a specified course in SPPU exam.
UA_AT_L3	30% students scoring more than 66% in a specified course in SPPU exam.

B. Measuring Course Outcomes attainment

UA_AC_AT_L1	Actual percentage of students scored more than 40% in a specified course.
UA_AC_AT_L2	Actual percentage of students scored more than 60% in a specified course.
UA_AC_AT_L3	Actual percentage of students scored more than 66% in a specified course.

UA_CO_AT = [((UA_AC_AT_L1/ UA_ AT_L1) * 1) +
((UA_AC_AT_L2/ UA_ AT_L2) * 2) +
((UA_AC_AT_L2/ UA_ AT_L2)* 3)] / 6

C. Attainment through Internal Assessment

MT_AT_L1	90% students scoring more than 40% in a specified course in Unit test/ Assignments/ Lab
MT_AT_L2	40% students scoring more than 60% in specified course in Unit test/ Assignments/ Lab.
MT_AT_L3	30% students scoring more than 66% in a specified course in Unit test/ Assignments/ Lab.

D. Measuring Course Outcome Attainment

MT_AC_AT_L1	**Actual percentage of students scored more than 40% in a specified course in Unit test/ Assignments/ Lab.**
MT_AC_AT_L2	Actual percentage of students scored more than 60% in a specified course in Unit test/ Assignments/ Lab.
MT_AC_AT_L3	Actual percentage of students scored more than 66% in a specified course in Unit test/ Assignments/ Lab.

MT_CO_AT = [(MT_AC_AT_L1/ MT_ AT_L1) * 1) +
((MT_AC_AT_L2/ MT_ AT_L2) * 2) +
((MT_AC_AT_L2/ UA_ AT_L2) * 3)] / 6

E. Final Attainment

Sr. No.	Assessment	Weightage	Attainment
1	University Examination(All Phases) UAL	70%	UA_CO_AT
2	Mid Term/Institute Level Unit Test I ,II, Assignments, Practicals	30%	MT_CO_AT

CO_AT= 0.70 * UA_CO_AT + 0.30 * MT_CO_AT

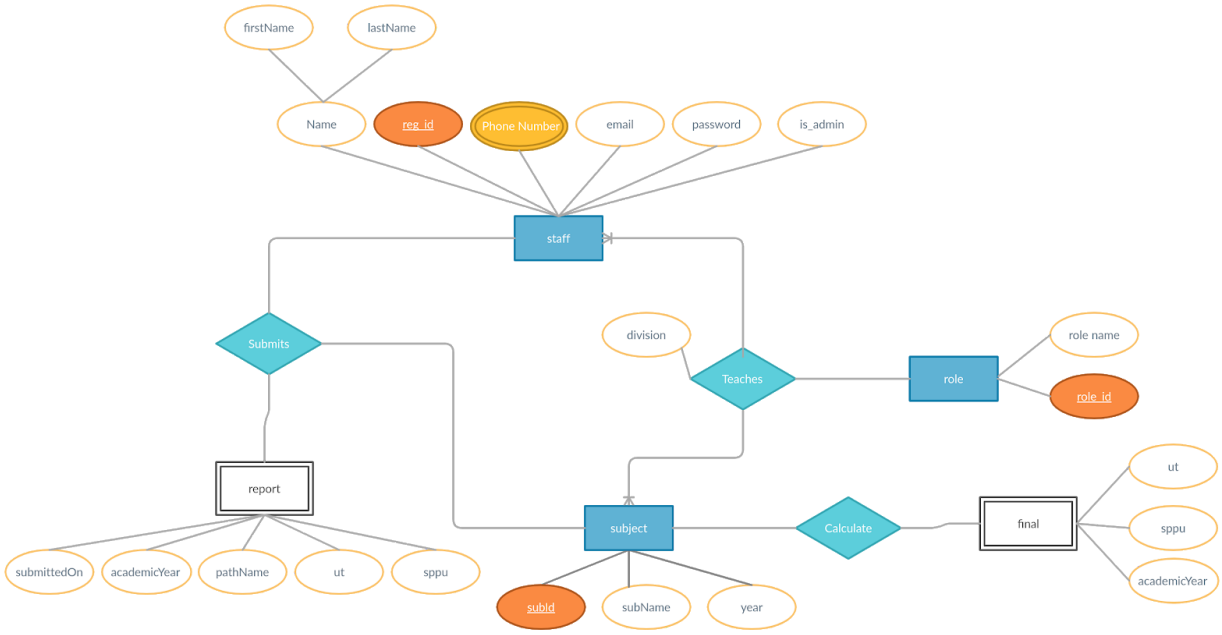
6.SCOPE

The scope of the project is virtually really wide as it can be slightly tweaked to fit custom use cases for other departments. The code is simple and readable and written in a systematic manner making it easier for developers to reuse the code.

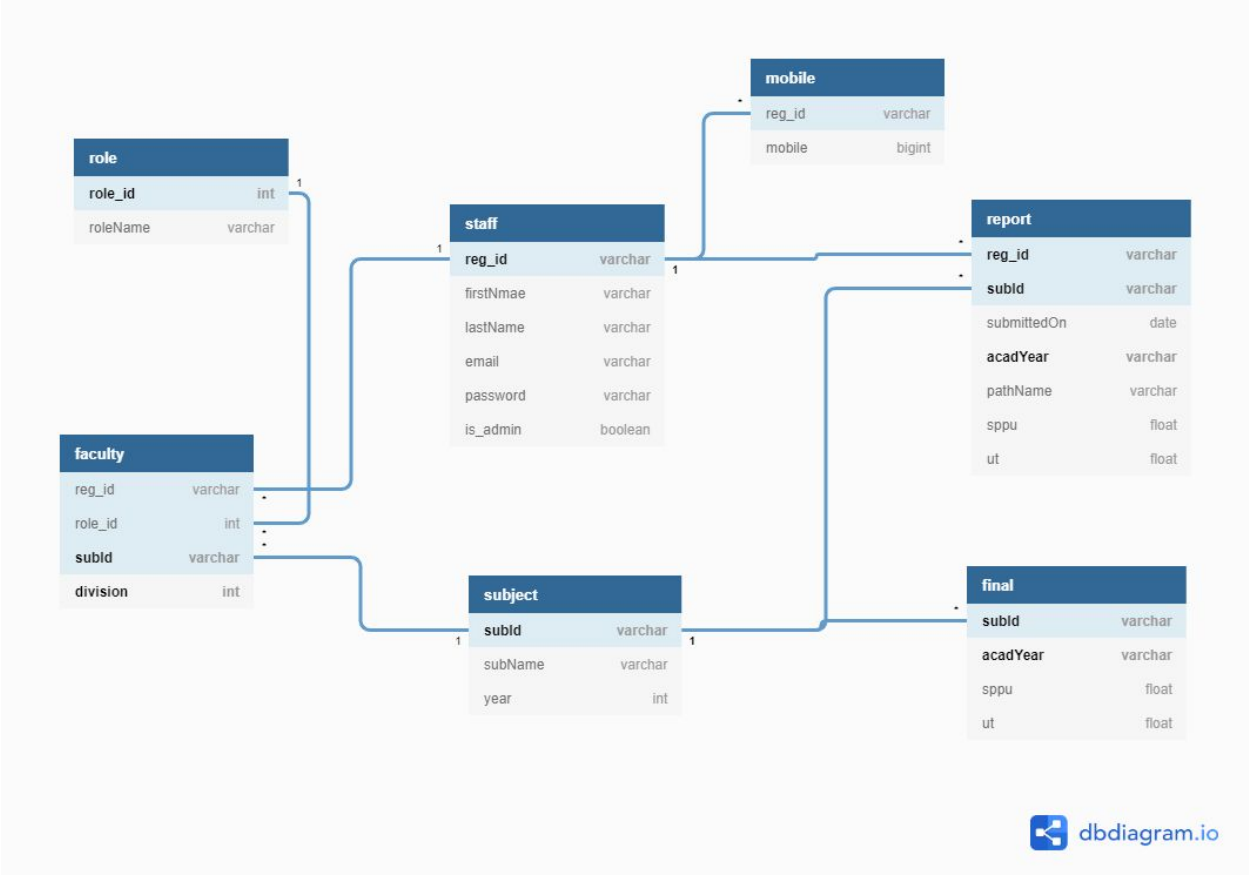
7.REQUIREMENTS:

- Ubuntu 18.04
- MySQL 8.0
- Node.js 12.0.0
- Node Package Manager(npm) 6.9.0
- Yarn 1.22.4

8.ER - Diagram


firstName
lastName
Name
reg_id
Phone Number
email
password
is_admin
staff
Submits
division
Teaches
role
role name
role_id
report
submittedOn
academicYear
pathName
ut
sppu
subject
subId
subName
year
Calculate
final
ut
sppu
academicYear

9.SCHEMA DIAGRAM



10.RELATIONAL DATABASE DESIGN

1. **Staff**

firstNam e	lastName	**reg_id**	email	is_admin	password

Attributes:

a. email - unique not null
b. firstName - varchar not null
c. is_admin - boolean default false
d. lastName - varchar not null
e. password - varchar
f. reg_id - varchar not null

2. **Faculty**

reg_id	role_id	**subId**	**division**

Attributes:

a. reg_id - varchar not null (foreign key)
b. role_id - int not null (foreign key)
c. subId - varchar not null (foreign key)
d. division - int not null

3. **Subject**

subId	subName	year

Attributes:

a. subId - varchar not null
b. subName - varchar not null
c. year - int not null

4. **Report**

reg_id	**subId**	submittedOn	**acadYear**	pathName	sppu	ut

Attributes:

a. **reg_id - varchar not null (foreign key)**
b. **subId - varchar not null (foreign key)**
c. submittedOn - date not null
d. **acadYear - varchar not null**
e. pathName - varchar not null
f. sppu - float
g. ut - float

5. **Final**

subId	**acadYear**	ut	sppu

Attributes:

a. **subId - varchar not null (foreign key)**
b. **acadYear - varchar not null**
c. ut - float not null
d. sppu - float not null

6. **Role**

role_id	roleName

Attributes:

a. **role_id - int not null**
b. roleName - varchar not null

7. **Mobile**

reg_id	mobileNo

Attributes:

a. reg_id - varchar not null (foreign key)
b. mobileNo - bigint not null

8. **Backup**

firstName	lastName	subject	submitted On	acadYear	pathName

Attributes:

a. firstName - varchar not null
b. lastName - varchar not null
c. subject - varchar not null
d. submittedOn - date not null
e. acadYear - varchar not null
f. pathName - varchar not null

DATABASE NORMALISATION:

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and update anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

First Normal Form:-
A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.
All the tables in the Target database have atomic fields and are hence in the First Normal Form (1NF). The separation of mobile number from staff into the mobile table makes the database compliant with 1NF.

Second Normal Form:
A relation is in the second normal form if it is in 1NF and every non-prime attribute of the relation is dependent on the whole of every candidate key.
No table in the Target database has any partial dependency and hence the database is in Second Normal Form(2NF). For example, in table report we cannot determine the non-prime attributes like pathName, submittedOn, ut and sppu by the proper subsets of the primary key, thus making it in 2nf as the table consists of no partial dependency.
The tables staff, role and subject consist of a single-attribute primary key so they are automatically in 2NF.

Third Normal Form:

Third normal form (3NF) is a database schema design approach for relational databases which uses normalizing principles to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management. A database relation is said to meet third normal form standards if it is already in the second normal form and all the attributes are functionally dependent on solely the primary key.

For example, in the staff table we have decomposed the table into two tables namely staff and role because if both tables would have been one, the reg_id of

staff would have been transitively dependent on the role name. Therefore we decompose the role table so as to avoid transitive dependency as well as avoid update anomalies.

12.GRAPHICAL USER INTERFACE

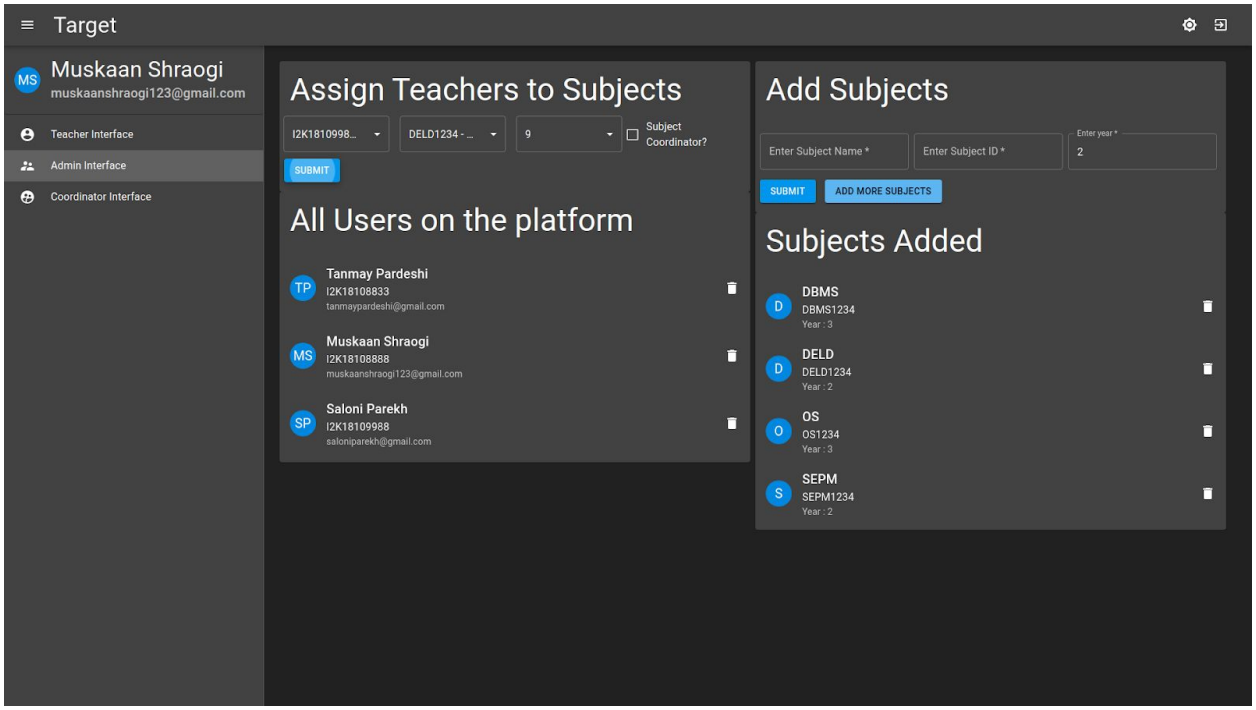
Target has a very user-friendly and an intuitive interface implemented in React to communicate with the User. To put in broad strokes, the platform majorly has two pages- The Login/SignUp Page and The Portal Page. The Portal Page would have it's own specific details depending upon the actor. Target also includes tool-tip hints to give a brief description of the particular input Field. Except this, we have also implemented a toggle switch which would alternate between the dark mode or light mode depending upon the users' preferences.

Features:

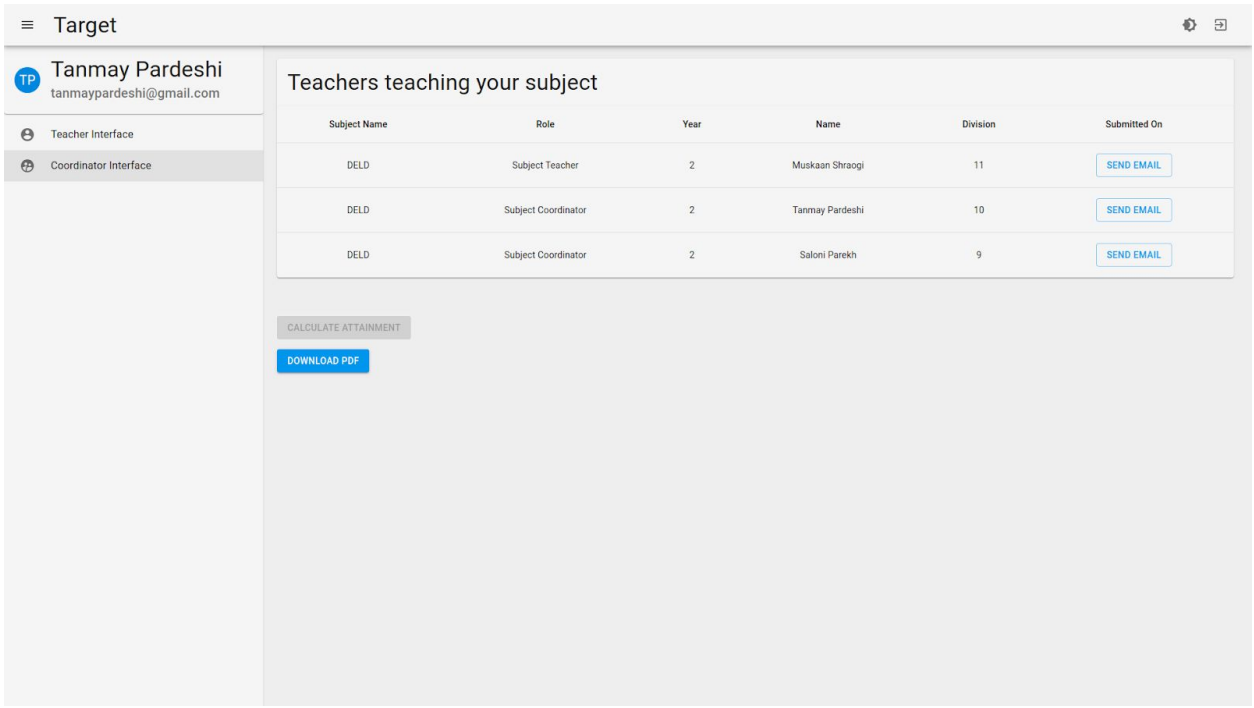
- Clean separation of various components to facilitate easy modification and revision.
- Depending upon the use-case the actor can use the following functionalities:
    - Admin: Add subjects to the platform and link them with their respective faculty members.
    - Faculty Member: Upload the respective subject excel sheets.
    - Subject Coordinator: Calculate the attainment (Once all the faculty members for that particular subject have uploaded their excel sheets)
- Subject Coordinator would also have an added functionality to send a reminder mail to the faculty member who hasn't uploaded the sheet via our platform.
- The Subject Coordinator can also download the attainment report in the pdf format.

Snapshots of the application:

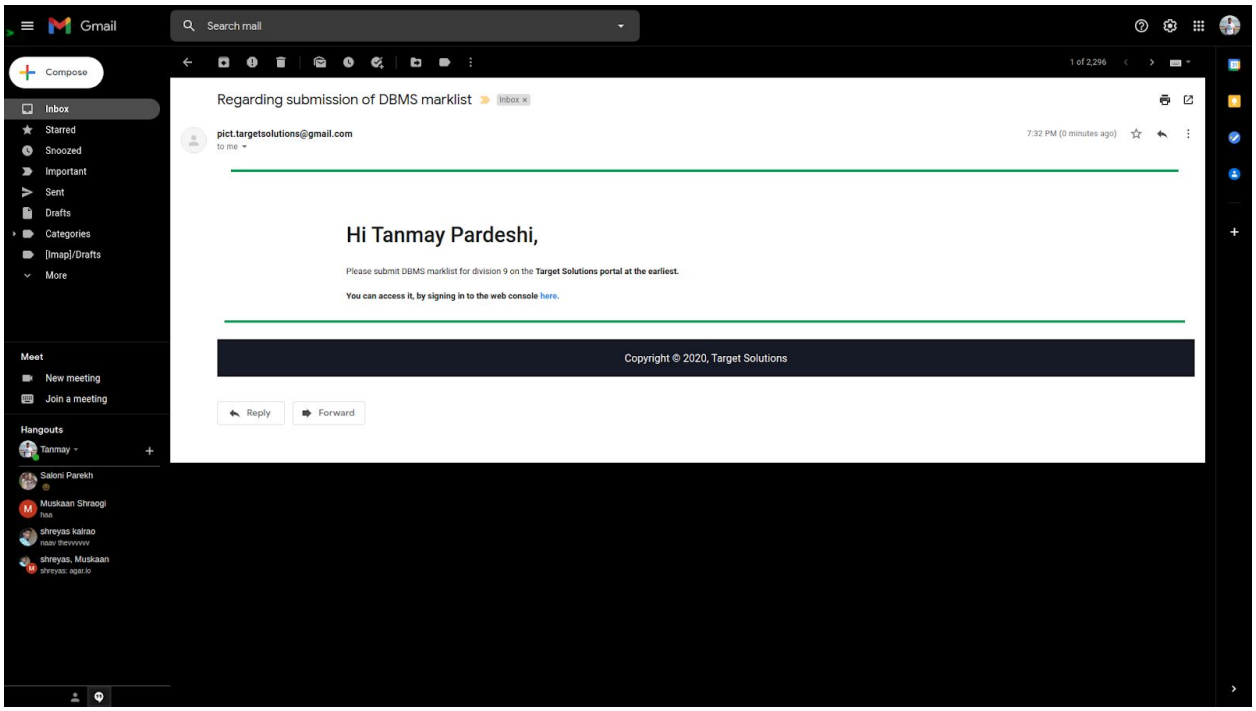
Admin Page:



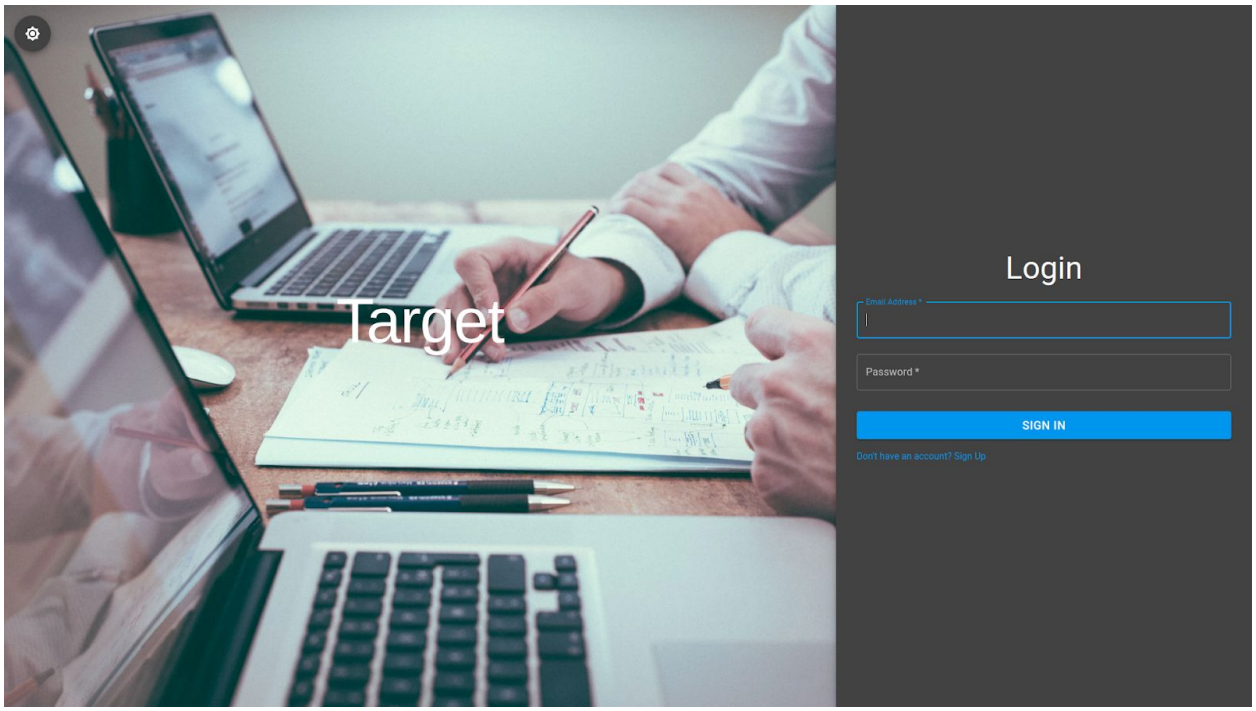
Coordinator Page:



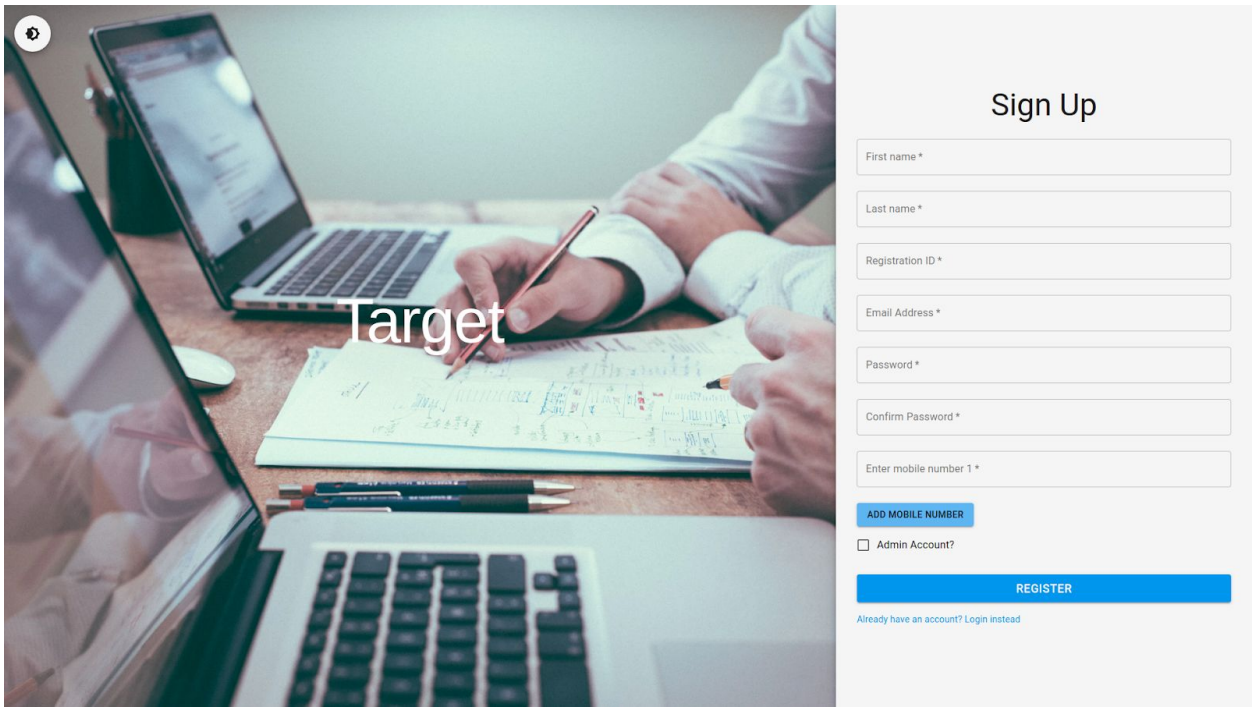
Reminder Mail:



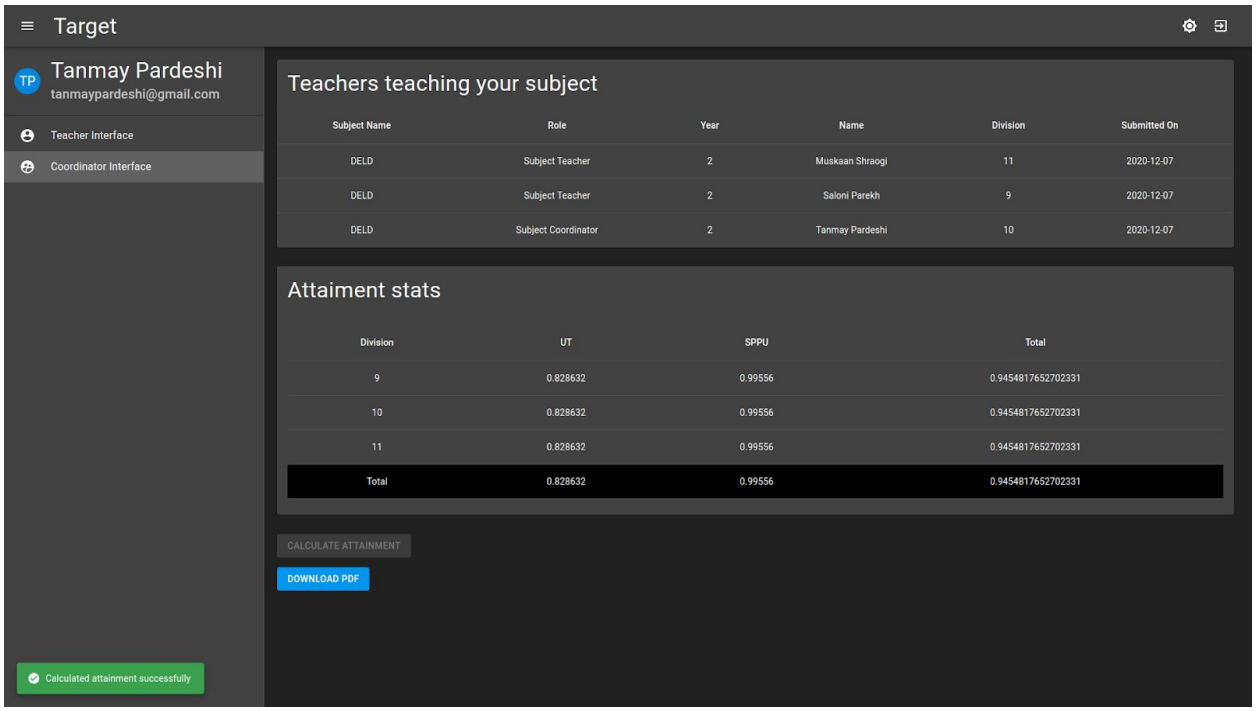
Login Page:


Target
Login
Email Address *
Password *
SIGN IN
Don't have an account? Sign Up

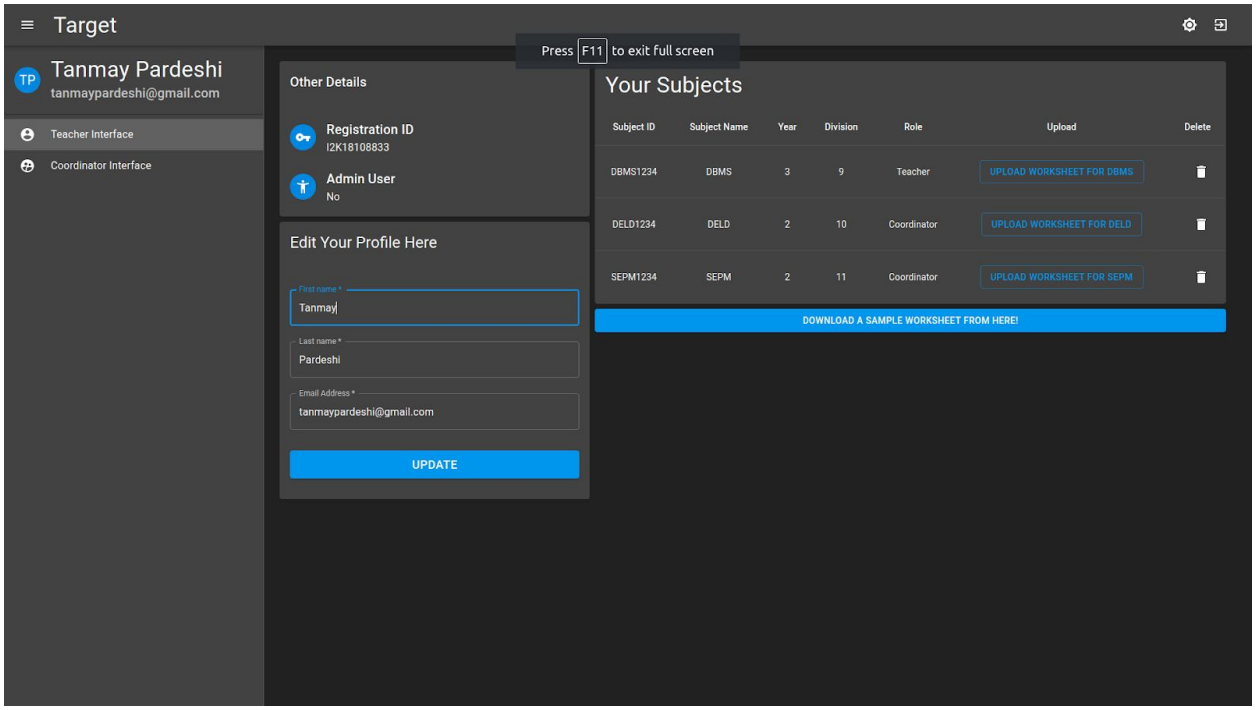
Register Page:



Result Dashboard:



Teacher Dashboard:



13.CONCLUSION:

We created a web based application for automated attainment generation and in the process learned about the practical uses of relational databases and the framework required to create a web application with MySQL as backend.

Concepts Used:

- MySQL
- NodeJS
- React

Software Used:

- Visual Studio Code
- Mozilla Firefox Web Browser
- Ubuntu 18.04