BadeLog – An Alumni Directory

Group No. - 6

Roll. No.	Name	E-mail
B17003	Akshat Malviya	B17003@students.iitmandi.ac.in
B17050	Neeraj Sharma	B17050@students.iitmandi.ac.in
B17100	Rohit Agarwal	B17100@students.iitmandi.ac.in
B17106	Tushar Tyagi	B17106@students.iitmandi.ac.in

Information and Database Management System (CS309)

Indian Institute of Technology, Mandi

May 2019

Index

Introduction

Problem Statement

To Develop an alumni directory for our institute with several functionalities for the students, faculties and alumni like sorting w.r.t. Batch, Field of interest, companies, projects, etc. along with the other basic features like messaging.

AIM

To develop a state-of-the-art portal for exchanging details about the alumni. Such a portal will result in better connections and will provide with various opportunities for students, alumni and faculty to collaborate in an efficient way. Alumni can provide details of openings in their respective companies so that students can apply for internships and jobs there with a reference from that alumni.

Functionalities

Here's a detailed list of the functionalities we've been able to incorporate in the project:

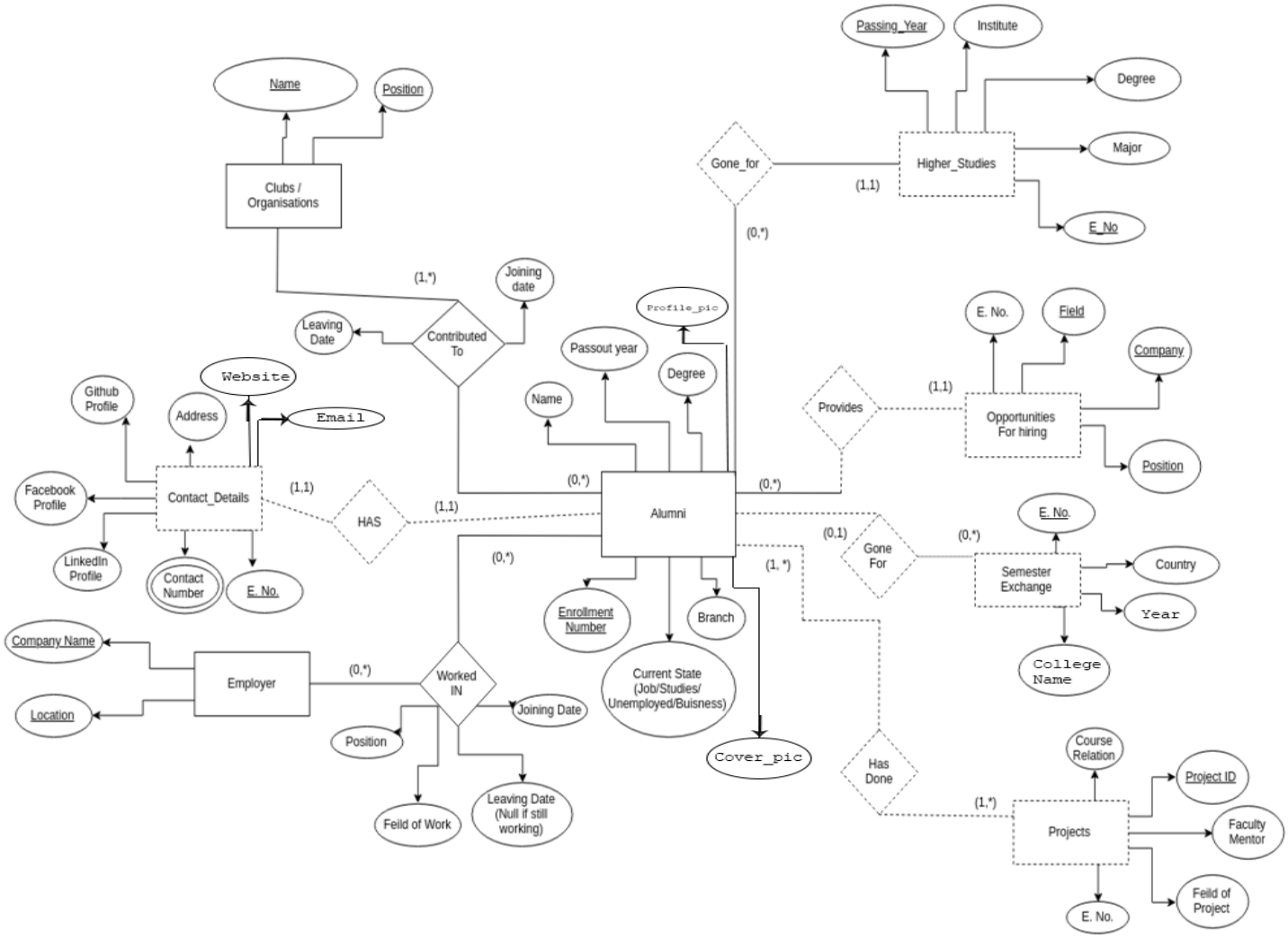
- Database of the alumnus students of our institute.
- Authentication of all users - **current students, alumni** and **faculty** as well!
- **Search, filter and sorting** functionality to search **for the alumni** based on different parameters - Passout year, branch, Current place of work, Field of work, Degree, Current company, etc.
- **Individual profile/page listing** details about each alumnus - containing details from his time in IIT Mandi to details about their current well being.
- **Connection platform** for the students (current with passed out ones)
- Options for **Alumni funding** for the institute, Student Gymkhana clubs.
- Web scraping to get the data of current student from insite.iitmandi.ac.in

All of the above points have been implemented in the project. Although, some are in a prototype.

Technologies Used

We used **MySQL** as Database because **MySQL** is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
Proper official documentation of MySQL helps us to understand the depth of the database. This also explains to us how RDBMS works.
We used Flask(Python web framework) with the jinja2 template for the Front end because It has a built-in development server and fast debugger. Flask also increases the performance as it is faster than other Frameworks like Django etc.
Other than this We also used flask_login to save the cookies of the user, flask_mail for Email authentication and flask_mysqldb for connecting flask with MySQL database.

Entity-Relation Diagram



Entities, Fields

Below is the list of Entities, their fields (which are listed below) and the corresponding function dependencies.

1. Alumni

	Attributes	Description
A	Enrollment No.	Roll Number of Alumni
B	Name	Name of Alumni
C	Passout year	Year in which Alumni graduated
D	Degree	Degree of Alumni
E	Branch	Branch of Alumni
F	Current Status	Current status like Job/Higher Studies/Business/Unemployed

2. Employer

A	Company Name	Any Industrial Company
B	Location	Location of Company

3. Clubs/Organizations

A	Name	Name of Clubs like Robotronics, STAC etc.
B	Position	Position in Club

4. Projects		
A	ProjectID	Unique ID of project
B	Faculty Mentor	Faculty associated with the project
C	Field of project	Field of project
D	Enrollment No.	Roll Number of Alumni
E	Course Relation	Course project or Extra project

5. Opportunities for Hiring		
A	Enrollment No.	Roll Number of Alumni
B	Field	Field of work
C	Company	Current company of Alumni
D	Position	Position in Company

6. Semester Exchange		
A	Enrollment No.	Roll Number of Alumni who applied for semester exchange
B	Country	Name of foreign Country
C	Year	Year in which an alumnus applied for it
D	College Name	Name of foreign colleges

7. Contact Details		
A	Enrollment No.	Roll Number of the user(Alumni)
B	Github Profile	Github link of the user
C	Facebook Profile	Facebook link of the user
D	LinkedIn Profile	LinkedIn link of the user
E	Contact Number	Contact number
F	Address	Address of user

8. Higher Studies		
A	Enrollment	Roll Number of the user(Alumni)
B	Major	Specialization field
C	Degree	Degree of Higher studies
D	Institute	Name of Institute
E	Passing year	Completion year of Degree

9. Worked_IN		
A	Enrollment No.	Roll Number of the user(Alumni)
B	Company Name	Current Company of user
C	Location	Current Location
D	Position	Position in company
E	Joining Date	Joining Date
F	Field of Work	Field of Work

G	Leaving Date	Null if the employer is still working

10. Contribute_To		
A	Enrollment No.	Roll Number of the user(Alumni)
B	Name	Name of user
C	Position	Position in Club
D	Leaving Date	Leaving date for the above position
E	Joining Date	Joining date for the above position

Constraints

1) **Alumni**: Primary Key (Enrollment number)
2) **Employer:** Primary Name(Company name, Location)
3) **Clubs/Organisation:** Primary Key (Name, Position)
4) **Projects:** Primary Key (ProjectID), Foreign Key (Enrollment number References Alumni)
5) **Opportunities for hiring:** Primary Key (Company, Position, Field), Foreign Key(Enrollment number References Alumni)
6) **Sem Exchange:** Primary Key (College name, Year), Foreign Key(Enrollment number References Alumni)
7) **Contact Details:** Primary Key (Enrollment number)
8) **Higher Studies:** Primary Key(Enrollment year, Passing year), Foreign Key(Enrollment number References Alumni)
9) **Worked_IN:** Primary Key(Enrollment number, company name, Location), Foreign Key(Enrollment number References Alumni, Company name References Employer, Location References Employer)
10) **Contributed_To:** Primary Key(Enrollment number, Position, Name), Foreign Key(Enrollment number References Alumni, Name References Clubs/Organisation, Position References Clubs/Organisation).

Functional Dependencies

1) **Alumni:** A → ABCDEF
2) **Employer:** AB → AB
3) **Clubs/Organisation:** AB → AB

4) **Projects:** A → ABCDE, DE → ABCDE
5) **Opportunities for Hiring:** BCD → ABCD
6) **Semester Exchange:** A → ABCD
7) **Contact Details:** A → ABCDEF
8) **Higher Studies:** AE → ABCDE
9) **Worked_IN:** ABC → ABCDEFG
10) **Contributed_To** : ABC → ABCDE

Size of the Database

The project database can have - Alumni database, Current students' database and Faculty database. This will contain data of 20 Alumni from the past four years.

Relation Schema

1. create table Employer(CompanyName VARCHAR(20),Location VARCHAR(20),PRIMARY KEY(CompanyName,Location));

2. create table Alumni(EnrollmentNumber VARCHAR(20),Name VARCHAR(25),PassoutYear int(5) ,Degree VARCHAR(20),Branch VARCHAR(10),CurrentState VARCHAR(25) ,Profile_image VARCHAR(40),Cover_image VARCHAR(40),PRIMARY KEY(EnrollmentNumber));

3. create table Clubs_organisations(Name VARCHAR(20),Position VARCHAR(30),PRIMARY KEY(Name,Position));

4. create table Contact_Details(EnrollmentNumber VARCHAR(20),Address VARCHAR(50),Facebook_profile VARCHAR(50),Github_profile VARCHAR(50),LinkedIn_profile VARCHAR(50),Contact_Number VARCHAR(11) ,Email VARCHAR(50), website_link VARCHAR(100), PRIMARY KEY(EnrollmentNumber));

5. create table Higher_Studies(EnrollmentNumber VARCHAR(20),Major VARCHAR(25),Degree VARCHAR(20) ,Institute VARCHAR(20),Passing_Year int(5),PRIMARY KEY(EnrollmentNumber,Passing_Year),Foreign Key(EnrollmentNumber)

References Alumni(EnrollmentNumber));

6. create table Opportunities_for_hiring(EnrollmentNumber VARCHAR(20),Field VARCHAR(50),Company VARCHAR(50),Position VARCHAR(50),PRIMARY KEY(Field,Company,Position),Foreign Key(EnrollmentNumber) References Alumni(EnrollmentNumber));

7. create table Semester_Exchange(EnrollmentNumber VARCHAR(20),Country VARCHAR(10),Year INT,College_name VARCHAR(20),PRIMARY KEY(EnrollmentNumber),Foreign Key(EnrollmentNumber)References Alumni(EnrollmentNumber));

8. create table Projects(Course_relation VARCHAR(25),ProjectID VARCHAR(10),Faculty_mentor VARCHAR(10),Field_of_project VARCHAR(10),EnrollmentNumber VARCHAR(20), PRIMARY KEY(ProjectID),Foreign Key(EnrollmentNumber) References Alumni(EnrollmentNumber));

9. create table Worked_In(EnrollmentNumber VARCHAR(20), CompanyName VARCHAR(20), Location VARCHAR(20), Position VARCHAR(50),JoiningDate VARCHAR(20),Field_of_work VARCHAR(50), LeavingDate VARCHAR(20),PRIMARY KEY(EnrollmentNumber, companyName, Location), Foreign Key(EnrollmentNumber) References Alumni(EnrollmentNumber) , Foreign Key(CompanyName, Location) References Employer(CompanyName, Location));

10. create table Contributed_To(EnrollmentNumber VARCHAR(20), Name VARCHAR(20), Position VARCHAR(30), JoiningDate VARCHAR(20), LeavingDate VARCHAR(20), Primary Key(EnrollmentNumber, Position, Name),
Foreign Key(EnrollmentNumber) References Alumni(EnrollmentNumber), Foreign Key(Name,Position) References
Clubs_organisations(Name,Position));

RDBMS Used

- ## Views

  Views are created for security purpose of database because it restricts the user from viewing certain column and rows in the table mean by using **view** we can apply the restriction on accessing the particular rows and columns for a specific user.
  Views also **increase the performance** as the time needed for search query decreases.

Views of tables -

Alumni Table -

1. For Degree Attribute
    - Degree_Btech : Enrollment Number for Degree = “B.Tech”
    - Degree_MTech : Enrollment Number for Degree = “M.Tech”
    - Degree_Phd : Enrollment Number for Degree = “Phd”

2. For Branch Attribute
    - Branch_CSE : Enrollment Number for Branch= “CSE”
    - Branch_EE: Enrollment Number for Branch= “EE”
    - Branch_ME: Enrollment Number for Branch = “ME”
    - Branch_CE: Enrollment Number for Branch = “CE”

3. For PassoutYear Attribute
    - PassoutYear_2019 : Enrollment Number for passout_year= 2019
    - PassoutYear_2018 : Enrollment Number for passout_year= 2018
    - PassoutYear_2017: Enrollment Number for passout_year= 2017
    - PassoutYear_2016 : Enrollment Number for passout_year= 2016

4. For Company and Location Attributes
    - Comp_Location_GS : Enrollment Number for Company = “Goldman” AND Location = “Bangalore”
    - Comp_Location_Google : Enrollment Number for Company = “Google” AND Location = “California”
    - Comp_Location_Siemens : Enrollment Number for Company = “Siemens” AND Location = “Bangalore”
    - Comp_Location_Microsoft : Enrollment Number for Company = “Microsoft” AND Location = “Hyderabad”

Opportunities_for_hiring Table -

4. For Field Attribute
    - Opportunities_Android: Enrollment Number for Field = “Android”
    - Opportunities_IOS: Enrollment Number for Field = “IOS”
    - Opportunities_WebD: Enrollment Number for Field = “WebD”

5. For Company and Position Attributes
    - Opportunities_ Google_DL: Enrollment Number for Company = “Google” AND Position = “Department Lead”
    - Opportunities_Ubisoft_JE: Enrollment Number for Company = “Ubisoft” AND Position = “Junior engine engineer”

- Opportunities_Siemens_DL : Enrollment Number for Company = “Siemens” AND Position = “Assistant DL Researcher”
- Opportunities_Zomato_dev : Enrollment Number for Company = “Zomato” AND Position = “Front end developer”

NOTE: Similarly Views are created for other tables also on different attributes depending on the requirements.

● Foreign Key Constraint

Various tables have a foreign key reference to the primary key of parent table **Alumni**.

Various foreign key constraints are maintained in it :

- Data of Alumni table is not allowed to delete So ON DELETE CASCADE is not used in the foreign table.
- Alumni table is the parent table so all other tables depend on it.
- Entries of child table are allowed to delete.

Foreign Key constraint will take care of the integrity of the database that we used.

● Indexes

Indexes are used to speed up the performance of queries by reducing the number of pages or rows.
Various indexes are used in it.
CREATE INDEX index_year ON Alumni (passoutYear)
CREATE INDEX index_Hiring ON Opportunities for hiring (Company ,Location)

MySQL Queries

Insert :

Insert query is used to insert the data into various tables detailed above.
Insert queries are used to insert the data into the database in the following cases :

- Signup of new user
- Details of alumni are added

The data can be inserted manually as well as through Web Scraping.

Update :

Update queries are used to update the database in the following cases when :

- Any user changes the password.
- Any user edits their profile details.
- Some alumni enter new content in hiring opportunities.
- If data is scrapped and needs to updated in the database.
- To store new messages in the database.

Search :

Select queries are used for searching the alumni upon various fields like Passout year, Branch, Degree, Current state(Job/Higher studies/Business/Unemployed), Current company, Location, Position, Field of work, Projects did, semester exchange etc.
Students can also use this feature to find the hiring opportunities provided by various alumni.

Security

- Use of HTTPS over HTTP
- Password Hashing
- Login Authentication
- Access with only IIT Mandi webmail

References

- As we want our platform to be hosted on IIT Mandi's server, we have hosted the database on a remote server. The database can be accessed through the following credentials:
    1. MYSQL_HOST: http://sql12.freemysqlhosting.net/
    2. MYSQL_USER: sql12292091
    3. MYSQL_PASSWORD: Neeraj@mysql
    4. MYSQL_DB: sql12292091

- MySQL official documentation: https://dev.mysql.com/doc/
- Flask official documentation: http://flask.pocoo.org/docs/1.0/
- Flask_login official documentation: https://flask-login.readthedocs.io/en/latest/
- Flask_webmail official documentation: https://pythonhosted.org/Flask-Mail/
- Flask_mysqldb official documentaion: https://flask-mysqldb.readthedocs.io/en/latest/

Experience

It was a very great learning experience to work on this project. The most important thing we learned was how to implement a database for a fully fledged application.

- It is now clear that to work with a database, we need to have a thorough knowledge of entity-relations, various constraints, functional dependencies.
- We learned that ***deciding the database*** for our applications depends upon various aspects like the size of the data, the demand for data, the requirement of administration, etc.
- We learned about various ***security protocols*** we need to ensure to make a robust application.
- We learned how to use SQL commands in a more **optimal way** than we used to do before.
- We learned about **r*emote hosting of databases.***
- We also learned about other technologies like ***Flask*** and ***Bootstrap*** which helped us in making the web interface of the application.
- We learned ***web scrapping*** for collecting data from various websites like IIT Mandi directory.

Overall it helped us to increase our **thinking capabilities** and gave us a wholesome experience of creating a production level application.