Payroll Automation System

Syracuse University, NY.

Tanay Pardeshi

Abstract: The project aims at automating the payroll process at the Comptroller's Office at Syracuse University. The major emphasis of this project is to replace the existing paper-based payroll tracking system with an automated database management system to simplify the entire process of payroll by eliminating the presence of manual labor.

School of Information Studies
SYRACUSE UNIVERSITY

Table of Contents:

Project Summary:

Background:

The Payroll Office at the Comptroller's Office at Syracuse University is responsible for handling payroll related operations. To obtain employment at the University, a student has to first approach prospective employers personally and apply for job positions. After applying, the student receives an email update on the availability of the position. On acquiring the position, the student receives an I-9 form from the University's HR Office. The I9 form must be handed over to the employer for verification. After the verification is complete, the employer provides an employment letter to the student. The final step is to obtain a Social Security Number from the Social Security Administration by providing their unique ID number (SUID).

The existing system at the Payroll Office keeps track of the records using a paper based format. This is problematic since keeping track of paper based records is an arduous task. Moreover, the student employees need to visit several places personally to complete the entire process which involves a lot of a manual labor. Hence, to automate this situation, I propose a database system that keeps track of the payroll process of the student employees and excludes the need for manual labor.

Proposed Solution:

- Student applies for jobs via an online job portal to seek employment.
- Employers can approve or reject an application through the same job portal.
- Once accepted, the employer sends out approval notices to prospective student employees.
- The students can then send the employment approval email to the HR office to get the I9 from.
- The HR office sends the I9 form electronically to the students.
- The student provides his electronic signature and sends the form over to the designated employer.
- Employer will verify the form and provide his electronic signature on the I9 form and send an employment letter to the student via email.
- The student applies to the Social Security Administration for receiving his Social Security Number.

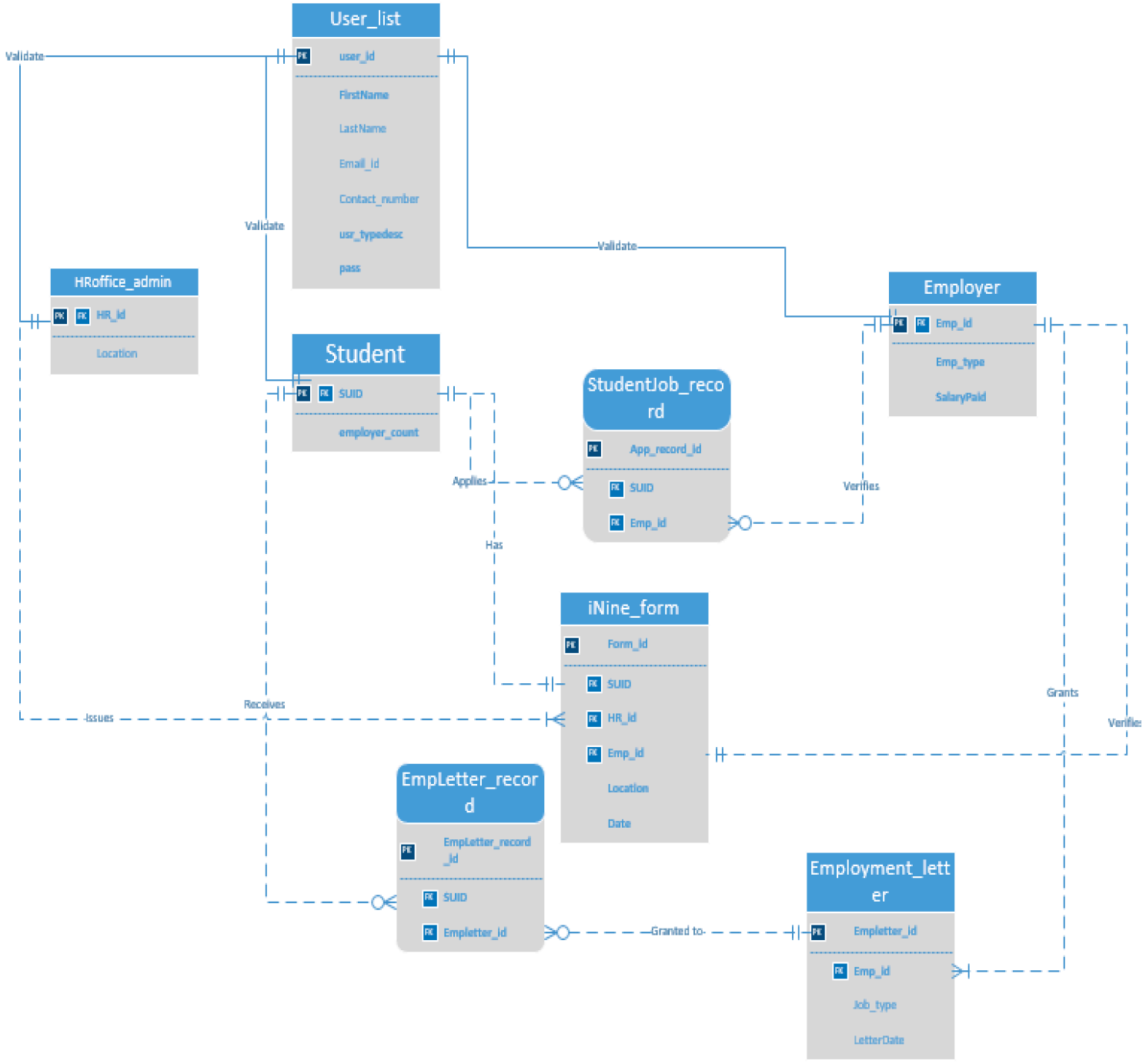
Tables and Attributes:

Database: **Payroll_automation**	**This database contains all the tables and relations that together build the Payroll Automation System.**
User_list	**Contains all types of user information**
user_id FirstName LastName Email_id Contact_number user_typedesc pass	PRIMARY KEY: Each user has a unique user id.
Student	**Child entity of User_list. Contains information about student applicants retrieved from the User_list table.**
SUID employer_count	PRIMARY KEY: Each user has a unique user_id. Each student has a unique SUID. FOREIGN KEY: Associated with primary key of 'User_list' table.
HRoffice_admin	**Child entity of User_list. Contains information about the HR office Administrator retrieved from the User_list table.**
HR_id Location	PRIMARY KEY: Each user has a unique user_id. The HR office Administrator has a unique HR_id. FOREIGN KEY: Associated with primary key of 'User_list' table.
Employer	**Child entity of User_list. Contains employer specific information retrieved from the User_list table.**

Emp_id Emp_type SalaryPaid	PRIMARY KEY: Each user has a unique user_id. Emp_id is unique for each employer. FOREIGN KEY: Associated with primary key of 'User_list' table.
StudentJob_record	**Associative entity that keeps track of the student job application details.**
App_record_id SUID Emp_id	PRIMARY KEY: Auto incremented integer identifying each record. FOREIGN KEY: SUID references the unique student id from Student table. Emp_id references the unique employer id from the Employer table.
iNine_form	**This entity is used to store critical information about the I9 form granted to a student after employment.**
Form_id SUID HR_id Emp_id Location Date	PRIMARY KEY: Form_id is an identification number that uniquely identifies a student's I9 information. FOREIGN KEY: SUID references the unique student id from Student table. Emp_id references the unique employer id from the Employer table. HR_id references the unique HR id from the HRoffice_admin table.
EmpLetter_record	**Associative entity to keep track of employment letter records sent to Students.**
EmpLetter_record_id SUID Empletter_id	PRIMARY KEY: Auto incremented integer identifying each record. FOREIGN KEY: SUID references the unique student id from Student table. Emp_id references the unique employer id from the Employer table.

Employment_letter	**Entity that is used to store information on employment letters.**
Empletter_id Emp_id Job_type LetterDate	PRIMARY KEY: Empletter_id is a unique identification number for an employment letter. FOREIGN KEY: Emp_id references the unique employer id from the Employer table.

Entity Relationship Diagram: MS Visio



Business Rules:

- The primary users of the system are Students, Employers and HR Office Administrators.
- A student can enroll as a new user to the system by giving his unique credentials.
- A student can receive multiple employment letters.
- A student is granted one and only one I9 form irrespective of the number of employment offers granted to the student.
- An employer can have more than one student employees working under him.
- The HR office Admin can verify a student employee's I9 form.
- The HR office Admin generates only one I9 form per student.

Database Infrastructure:

The database infrastructure is based on a client-server model. SQL server is used as the database engine and MS Access is used as the interface design tool. The database architecture is designed in MS Visio. Data is inserted, deleted, updated and queried from the SQL server database with the help of forms on Access. Useful data stored on SQL database can also be viewed with the help of reports generated through access.

SQL scripts for creating and inserting sample data:

Create Table queries:

User_list Table:

```
CREATE TABLE User_list
(
    user_id INTEGER primary key NOT NULL,
    FirstName VARCHAR(30) NOT NULL,
    LastName VARCHAR(30),
    Email_id VARCHAR(20),
    Contact_number INTEGER ,
    usertype_desc VARCHAR(20) NOT NULL,
    pass VARCHAR(30) NOT NULL
);
```

HRoffice_admin Table:

```
CREATE TABLE HRoffice_admin
(
    HR_id INTEGER NOT NULL,
    Location VARCHAR(20),
    CONSTRAINT HRoffice_admin_PK PRIMARY KEY(HR_id),
    CONSTRAINT HRoffice_admin_FK FOREIGN KEY(HR_id) REFERENCES User_list(user_id),
);
```

Student Table:

```
CREATE TABLE Student
(
    SUID INTEGER NOT NULL,
    employer_count INTEGER NOT NULL,
    CONSTRAINT Student_PK PRIMARY KEY(SUID),
    CONSTRAINT Student_FK FOREIGN KEY(SUID) REFERENCES User_list(user_id),
);
```

Employer Table:

```
CREATE TABLE Employer
(
    Emp_id INTEGER NOT NULL,
    Emp_type VARCHAR(30 )NOT NULL,
    SalaryPaid INTEGER NOT NULL,
    CONSTRAINT Employer_PK PRIMARY KEY(Emp_id),
    CONSTRAINT Employer_FK FOREIGN KEY(Emp_id) REFERENCES User_list(user_id),
);
```

iNine_form Table:

```
CREATE TABLE iNine_form
(
    Form_id INTEGER NOT NULL,
    SUID INTEGER NOT NULL,
    HR_id INTEGER NOT NULL,
    Emp_id INTEGER NOT NULL,
    Location VARCHAR(30)NOT NULL,
    Date DATE NOT NULL,
    CONSTRAINT iNine_form_PK PRIMARY KEY(Form_id),
    CONSTRAINT iNine_form_FK1 FOREIGN KEY(SUID) REFERENCES Student(SUID),
    CONSTRAINT iNine_form_FK2 FOREIGN KEY(HR_id) REFERENCES HRoffice_admin(HR_id),
    CONSTRAINT iNine_form_FK3 FOREIGN KEY(Emp_id) REFERENCES Employer(Emp_id)
);
```

Employment_letter Table:

```
CREATE TABLE Employment_letter
(
    Empletter_id INTEGER NOT NULL,
    Emp_id INTEGER NOT NULL,
    Job_type VARCHAR(30),
    LetterDate DATE,
    CONSTRAINT Employment_letter_PK PRIMARY KEY(Empletter_id),
    CONSTRAINT Employment_letter_FK1 FOREIGN KEY(Emp_id) REFERENCES Employer(Emp_id),
);
```

StudentJob_record Table:

```
CREATE TABLE StudentJob_record
(
    App_record_id Integer NOT NULL,
    SUID INTEGER NOT NULL,
    Emp_id INTEGER NOT NULL,
    CONSTRAINT SJR_PK PRIMARY KEY(App_record_id),
    CONSTRAINT SJR_FK FOREIGN KEY(Emp_id) REFERENCES Employer(Emp_id),
    CONSTRAINT SJR_FK2 FOREIGN KEY(SUID) REFERENCES Student(SUID),
);
```

Empletter_record Table:

```
CREATE TABLE EmpLetter_record
(
    EmpLetter_record_id Integer NOT NULL,
    SUID INTEGER NOT NULL,
    Empletter_id INTEGER NOT NULL,
    CONSTRAINT EmpLetter_record_PK PRIMARY KEY(EmpLetter_record_id),
    CONSTRAINT EmpLetter_record_FK1 FOREIGN KEY(SUID) REFERENCES Student(SUID),
    CONSTRAINT EmpLetter_record_FK2 FOREIGN KEY(Empletter_id) REFERENCES Employment_letter(Empletter_id)
);
```

Insert Queries:

User_list Table:

```
/*Inserting data in User_list table */f
INSERT INTO User_list VALUES(100, 'Rahul', 'Dravid','rd@syr.edu', 123213,'hr admin','rahul');

INSERT INTO User_list VALUES(200, 'Sachin', 'Tendulkar','st@syr.edu', 132411,'Student','sachin');
INSERT INTO User_list VALUES(201, 'Saurav', 'Ganguly','sg@syr.edu',412311, 'Student','saurav');
INSERT INTO User_list VALUES(202, 'VVS', 'Laxman','vvsl@syr.edu',132411, 'Student','vvs');
INSERT INTO User_list VALUES(203, 'KL', 'Rahul','kr@syr.edu',132431, 'Student','klr');
INSERT INTO User_list VALUES(204, 'Shikhar', 'Dhawan','sd@syr.edu',132811, 'Student','shikhar');
INSERT INTO User_list VALUES(205, 'Cheteshwar', 'Pujara','cp@syr.edu',132441, 'Student','pujara');
INSERT INTO User_list VALUES(206, 'Cristiano', 'Ronaldo','cr7@syr.edu',322411, 'Student','cristiano');
INSERT INTO User_list VALUES(207, 'Lionel', 'Messi','lm@syr.edu',132423, 'Student','messi');
INSERT INTO User_list VALUES(208, 'David', 'Beckham','db@syr.edu',232411, 'Student','david');
INSERT INTO User_list VALUES(209, 'Suresh', 'Raina','sr@syr.edu',431231, 'Student','suresh');

INSERT INTO User_list VALUES(300, 'Ashish', 'Nehra','an@syr.edu',312412, 'employer','ashish');
INSERT INTO User_list VALUES(301, 'Umesh', 'Yadav','uy@syr.edu', 141242,'employer','umesh');
INSERT INTO User_list VALUES(302, 'Zaheer', 'Khan','zk@syr.edu',124241, 'employer','zaheer');
INSERT INTO User_list VALUES(303, 'David', 'Warner','dw@syr.edu', 134222,'employer','warner');
INSERT INTO User_list VALUES(304, 'Tim', 'Cook','tc@syr.edu', 131242,'employer','tim');
INSERT INTO User_list VALUES(305, 'Steve', 'Jobbs','sj@syr.edu', 134247,'employer','steve');
INSERT INTO User_list VALUES(306, 'Bill', 'Gates','bg@syr.edu', 183242,'employer','bill');
INSERT INTO User_list VALUES(307, 'Andrew', 'Symonds','as@syr.edu', 134242,'employer','andrew');
INSERT INTO User_list VALUES(308, 'Brett', 'Lee','bl@syr.edu', 134242,'employer','brett');
INSERT INTO User_list VALUES(309, 'Virat', 'Kohli','vk@syr.edu', 134242,'employer','virat');
```

Result:

	user_id	FirstName	LastName	Email_id	Contact_number	usertype_desc	pass
1	100	Rahul	Dravid	rd@syr.edu	123213	hr admin	rahul
2	200	Sachin	Tendulkar	st@syr.edu	132411	Student	sachin
3	201	Saurav	Ganguly	sg@syr.edu	412311	Student	saurav
4	202	VVS	Laxman	vvsl@syr.edu	132411	Student	vvs
5	203	KL	Rahul	kr@syr.edu	132431	Student	klr
6	204	Shikhar	Dhawan	sd@syr.edu	132811	Student	shikhar
7	205	Cheteshwar	Pujara	cp@syr.edu	132441	Student	pujara
8	206	Cristiano	Ronaldo	cr7@syr.edu	322411	Student	cristiano
9	207	Lionel	Messi	lm@syr.edu	132423	Student	messi
10	208	David	Beckham	db@syr.edu	232411	Student	david
11	209	Suresh	Raina	sr@syr.edu	431231	Student	suresh
12	300	Ashish	Nehra	an@syr.edu	312412	employer	ashish
13	301	Umesh	Yadav	uy@syr.edu	141242	employer	umesh
14	302	Zaheer	Khan	zk@syr.edu	124241	employer	zaheer
15	303	David	Warner	dw@syr.edu	134222	employer	warner
16	304	Tim	Cook	tc@syr.edu	131242	employer	tim
17	305	Steve	Jobbs	sj@syr.edu	134247	employer	steve
18	306	Bill	Gates	bg@syr.edu	183242	employer	bill
19	307	Andrew	Symonds	as@syr.edu	134242	employer	andrew
20	308	Brett	Lee	bl@syr.edu	134242	employer	brett
21	309	Virat	Kohli	vk@syr.edu	134242	employer	virat

HRoffice_admin Table:

```
/*Inserting data into HR office admin table*/
INSERT INTO HRoffice_admin VALUES(100, 'Steele Hall');
```

Result:

	HR_id	Location
1	100	Steele Hall

Student Table:

```
/*Inserting data into Student table*/
INSERT INTO Student VALUES(200,1);
INSERT INTO Student VALUES(201,1);
INSERT INTO Student VALUES(202,1);
INSERT INTO Student VALUES(203,1);
INSERT INTO Student VALUES(204,1);
INSERT INTO Student VALUES(205,1);
INSERT INTO Student VALUES(206,1);
INSERT INTO Student VALUES(207,1);
INSERT INTO Student VALUES(208,1);
INSERT INTO Student VALUES(209,1);
```

Result:

	SUID	employer_count
1	200	1
2	201	1
3	202	1
4	203	1
5	204	1
6	205	1
7	206	1
8	207	1
9	208	1
10	209	1

Employer Table:

```
/*Inserting data into Employer table*/
INSERT INTO Employer VALUES(300, 'Graham Hall', 10);
INSERT INTO Employer VALUES(301, 'Career Dome', 12);
INSERT INTO Employer VALUES(302, 'Ernie Davis Hall', 11);
INSERT INTO Employer VALUES(303, 'Flint Hall', 12);
INSERT INTO Employer VALUES(304, 'Flint Hall', 12);
INSERT INTO Employer VALUES(305, 'Graham Hall', 10);
INSERT INTO Employer VALUES(306, 'Flint Hall', 12);
INSERT INTO Employer VALUES(307, 'Ernie Davis Hall', 11);
INSERT INTO Employer VALUES(308, 'Flint Hall', 12);
INSERT INTO Employer VALUES(309, 'Ernie Davis Hall', 11);
```

Result:

	Emp_id	Emp_type	SalaryPaid
1	300	Graham Hall	10
2	301	Career Dome	12
3	302	Ernie Davis Hall	11
4	303	Flint Hall	12
5	304	Flint Hall	12
6	305	Graham Hall	10
7	306	Flint Hall	12
8	307	Ernie Davis Hall	11
9	308	Flint Hall	12
10	309	Ernie Davis Hall	11

iNine_form Table:

```
/*Inserting data into iNine Form table*/
INSERT INTO iNine_Form VALUES(1,200,100,300,'Steele Hall', '12/2/2010');
INSERT INTO iNine_Form VALUES(2,201,100,301,'Steele Hall', '12/3/2010');
INSERT INTO iNine_Form VALUES(3,202,100,302,'Steele Hall', '12/4/2010');
INSERT INTO iNine_Form VALUES(4,203,100,303,'Steele Hall', '12/5/2010');
INSERT INTO iNine_Form VALUES(5,204,100,304,'Comptroller Office', '12/5/2010');
INSERT INTO iNine_Form VALUES(6,205,100,305,'Steele Hall', '12/5/2010');
INSERT INTO iNine_Form VALUES(7,206,100,306,'Comptroller Office', '12/5/2010');
INSERT INTO iNine_Form VALUES(8,207,100,307,'Steele Hall', '12/5/2010');
INSERT INTO iNine_Form VALUES(9,208,100,308,'Steele Hall', '12/5/2010');
INSERT INTO iNine_Form VALUES(10,209,100,309,'Comptroller Office', '12/5/2010');
```

Result:

	Form_id	SUID	HR_id	Emp_id	Location	Date
1	1	200	100	300	Steele Hall	2010-12-02
2	2	201	100	301	Steele Hall	2010-12-03
3	3	202	100	302	Steele Hall	2010-12-04
4	4	203	100	303	Steele Hall	2010-12-05
5	5	204	100	304	Comptroller Office	2010-12-05
6	6	205	100	305	Steele Hall	2010-12-05
7	7	206	100	306	Comptroller Office	2010-12-05
8	8	207	100	307	Steele Hall	2010-12-05
9	9	208	100	308	Steele Hall	2010-12-05
10	10	209	100	309	Comptroller Office	2010-12-05

Employment_letter Table:

```
/*Inseting data into Employment letter table*/
INSERT INTO Employment_letter VALUES(1,300,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(2,301,'Supervisor','12/1/2010');
INSERT INTO Employment_letter VALUES(3,302,'Supervisor', '12/1/2010');
INSERT INTO Employment_letter VALUES(4,303,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(5,304,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(6,305,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(7,306,'Supervisor', '12/1/2010');
INSERT INTO Employment_letter VALUES(8,307,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(9,308,'Food services', '12/1/2010');
INSERT INTO Employment_letter VALUES(10,309,'Supervisor', '12/1/2010');
```

Result:

	Empletter_id	Emp_id	Job_type	LetterDate
1	1	300	Food services	2010-12-01
2	2	301	Supervisor	2010-12-01
3	3	302	Supervisor	2010-12-01
4	4	303	Food services	2010-12-01
5	5	304	Food services	2010-12-01
6	6	305	Food services	2010-12-01
7	7	306	Supervisor	2010-12-01
8	8	307	Food services	2010-12-01
9	9	308	Food services	2010-12-01
10	10	309	Supervisor	2010-12-01

StudentJob_record Table:

```
/*Inserting data into StudentJob_record table */
INSERT INTO StudentJob_record VALUES(1,200,300);
INSERT INTO StudentJob_record VALUES(2,201,301);
INSERT INTO StudentJob_record VALUES(3,202,302);
INSERT INTO StudentJob_record VALUES(4,203,303);
INSERT INTO StudentJob_record VALUES(5,204,304);
INSERT INTO StudentJob_record VALUES(6,205,305);
INSERT INTO StudentJob_record VALUES(7,206,306);
INSERT INTO StudentJob_record VALUES(8,207,307);
INSERT INTO StudentJob_record VALUES(9,208,308);
INSERT INTO StudentJob_record VALUES(10,209,309);
```

Result:

	App_record_id	SUID	Emp_id
1	1	200	300
2	2	201	301
3	3	202	302
4	4	203	303
5	5	204	304
6	6	205	305
7	7	206	306
8	8	207	307
9	9	208	308
10	10	209	309

Empletter_record Table:

```
/*Inserting data into Empletter_record table */
INSERT INTO EmpLetter_record VALUES(1,200,1)
INSERT INTO EmpLetter_record VALUES(2,201,2)
INSERT INTO EmpLetter_record VALUES(3,202,3)
INSERT INTO EmpLetter_record VALUES(4,203,4)
INSERT INTO EmpLetter_record VALUES(5,204,5)
INSERT INTO EmpLetter_record VALUES(6,205,6)
INSERT INTO EmpLetter_record VALUES(7,206,7)
INSERT INTO EmpLetter_record VALUES(8,207,8)
INSERT INTO EmpLetter_record VALUES(9,208,9)
INSERT INTO EmpLetter_record VALUES(10,209,10)
```

Result:

	EmpLetter_record_id	SUID	Empletter_id
1	1	200	1
2	2	201	2
3	3	202	3
4	4	203	4
5	5	204	5
6	6	205	6
7	7	206	7
8	8	207	8
9	9	208	9
10	10	209	10

Major Data Questions:

Users:

The different users involved in this system are as follows:

- Students: Sign up, Generate employer count, Calculate Total Income, Check employment letter status, Check I9 form status.

- Employers: Add student employee, Generate employment letter, Check I9 form validity.

- Human Resources Office Administrator: Generate I9 form, Verify I9 form.

Major data questions answered:

- How many employers is a student employee working under?
- What is the total income of a student? (Considering all the employers he works under)
- How many employment letters are issued to a student employee?
- Is the I9 form issued and verified by HR admin and employer respectively?
- Is the employment letter generated for a student employee?

Why Students query the database:

-Register as a new student:

A new student can sign up to the system with the help of his unique user id and password in order to gain access to the Payroll database system.

-Get information on Employers:

A student can work under more than one employers. The students can get information on the number of employers he works under with the click of a button.

-View status of Employment letter:

A student can query the database to view the employment status of the job positions he has applied to. A student can apply to more than one employer for more than one job position. Therefore, it becomes easy for the student to track his job offers from prospective employers.

-Generate total income:

As stated above, a student can work under multiple employers. Therefore, this system also allows the student employee to calculate the total income he gains under all the employers he works under. This helps in assessing the student's total salary.

-Check I9 form status:

Once the student has applied for a job, he must have his I9 form generated from the HR office administrator. Therefore, this feature allows the student to check the status of his I9 form i.e. he can check whether the I9 form is generated or not.

Why Employers query the database:

- Generate employment letter:

An employer can add a new Student employer for a job position after he/she has been approved for that positon. After adding the employee to the records, an employment letter must be

generated for that student. Therefore, the employer can generate an employment letter for the student to authorize his/her employment.

-Add student employee to the record:

An employer can add a new student employee under his records if he/she proves to be fit for the position. The employer can add multiple students to his records for different job positions.

-View I9 form validity of a Student employee:

In order to gain employment, a student must have an I9 form generated from the HR office. Therefore, to check the validity of an I9 form of a particular student employee, the employer has been given access to check the I9 form records of the student employee who wishes to seek employment under him.

Why HR Office Admin queries the database:

-Check I9 status of an employee:

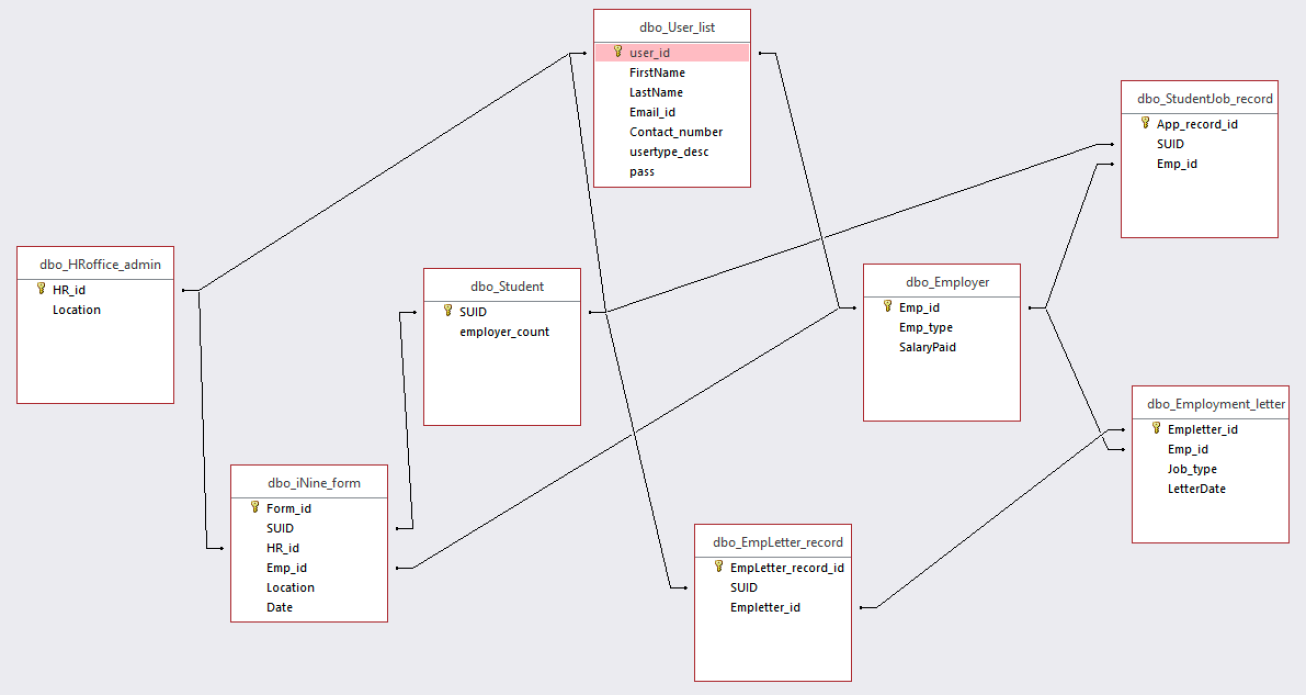
The HR Office Admin can check the status of a student employees I9 form based on his job status. The I9 form can be then granted to the student only if the student has successfully secured a job position under an employer.

-Generate an I9 form for an employee:

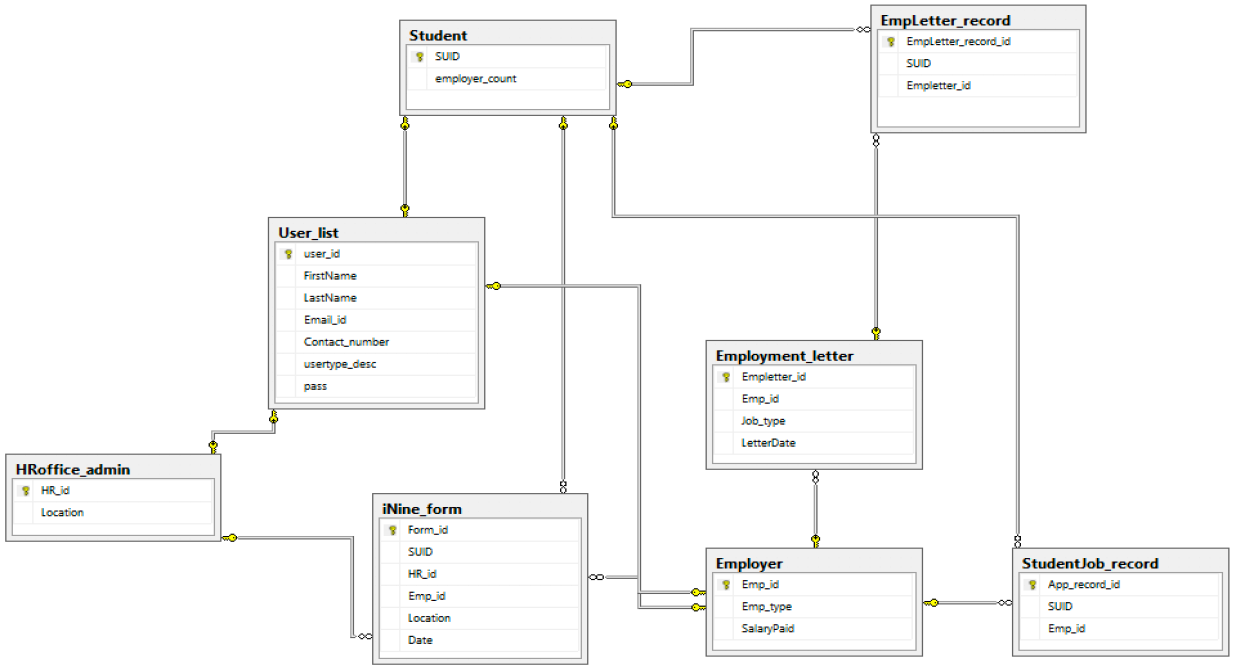
The HR office can generate an I9 form for a student employee who wishes to seek employment. However, one student is entitled to one I9 form only.

Relationship Diagrams:

MS Access Relationship Diagram:



SQL Diagram:



Forms and Reports:

The different users involved in this system are as follows:

- Students:

- Employers:

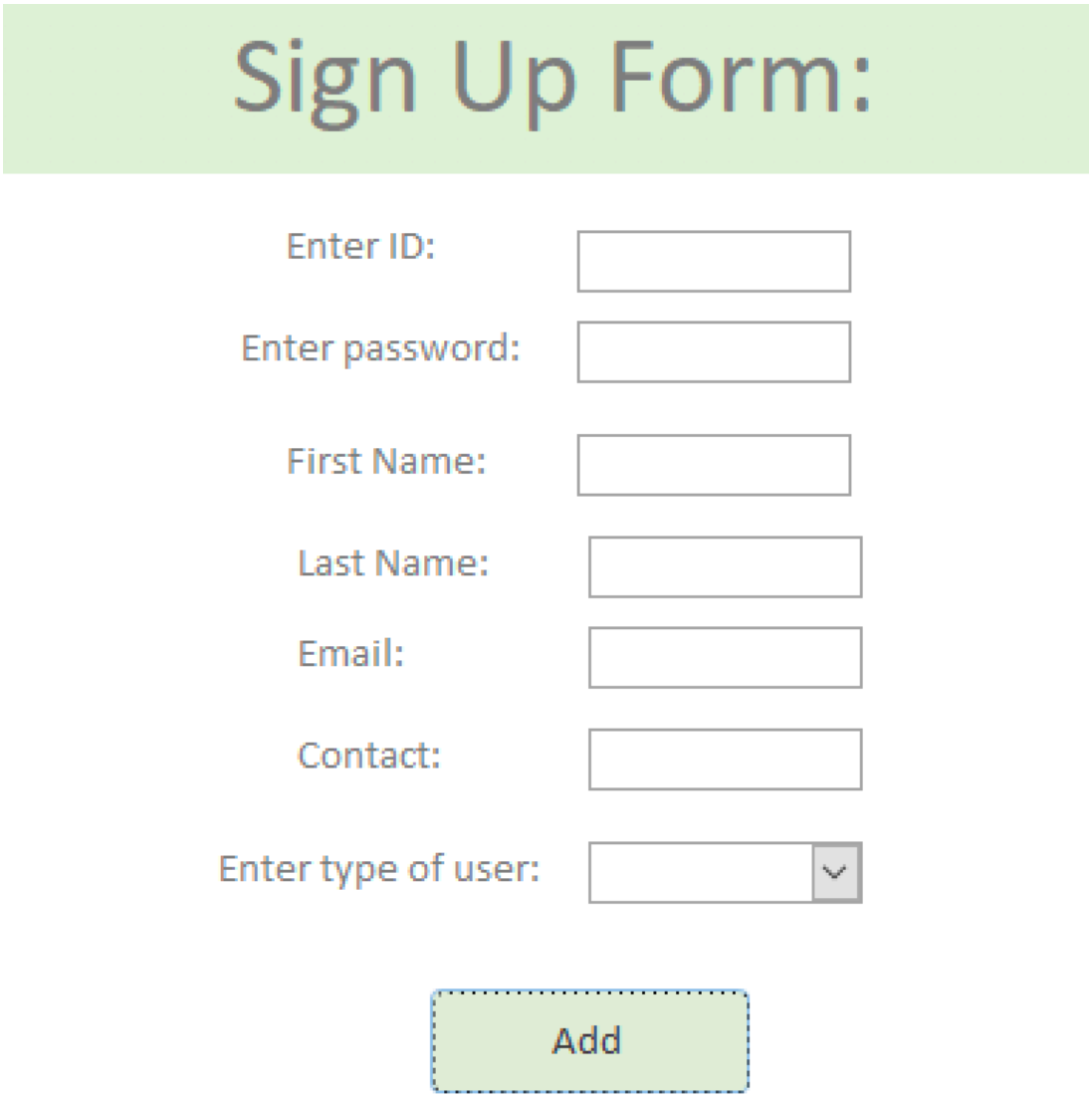
- Human Resources Office Administrator:

Login Page:

The user enters his credentials and is granted access to either the Student Page, Employer Page or HR Admin page based on the user id and password details.

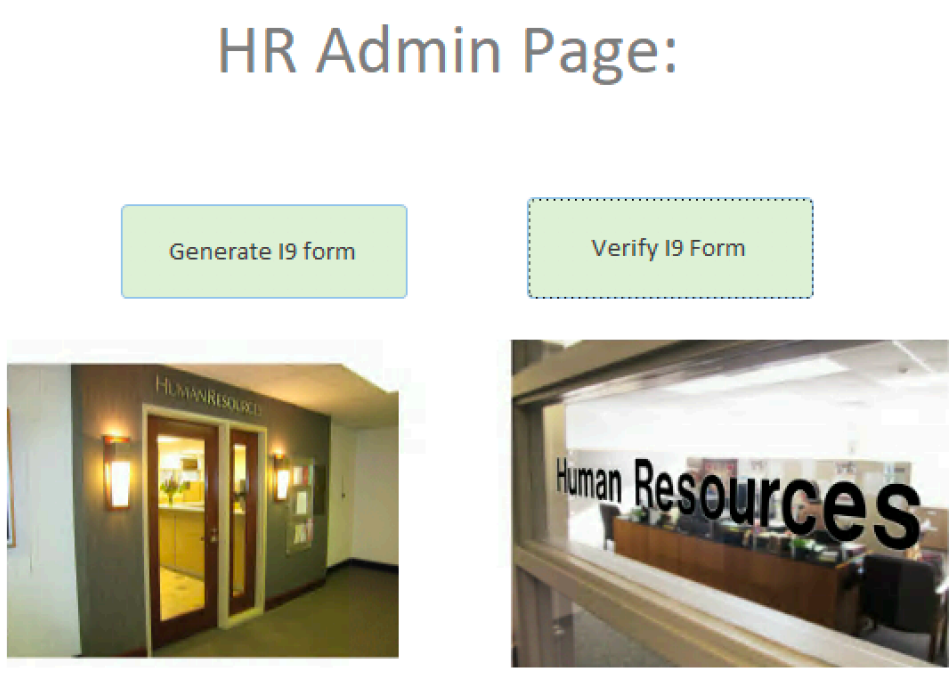
Sign Up Form:

A new student can register himself in the system by entering unique details such as user id and password.



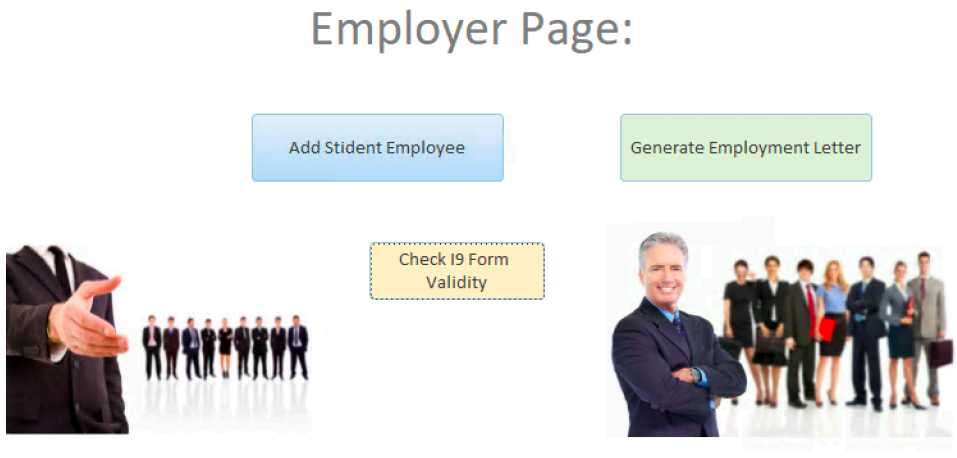
HR Admin Form:

This page is exclusively for the HR office Administrators and allows them to either generate a new I9 form or Verify existing I9 form records.

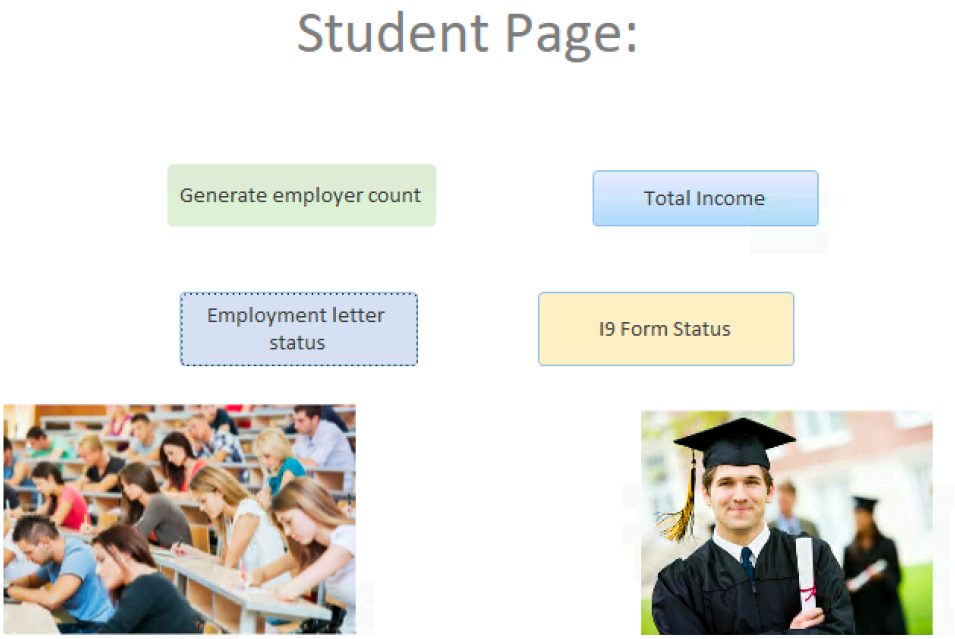


Employer Form:

This page is exclusively for the Employers and allows them to add a new student employee to the record, generate employment letters and check I9 form validity of the employees.

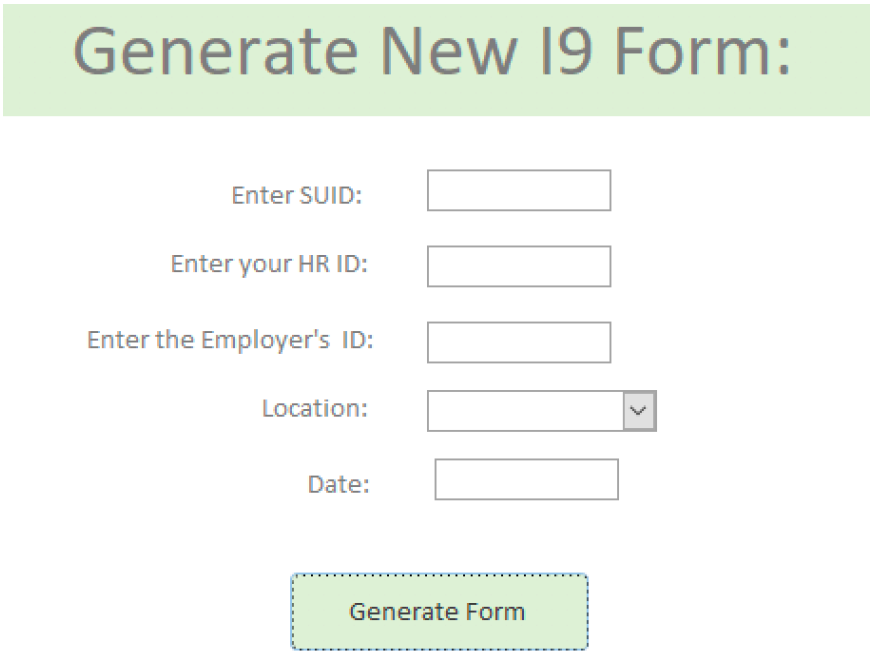


Student Form:



HR Admin → Generate I9 Form

This form allows the HR admin to generate a new I9 form for an employee.



HR Admin → Verify I9 Form

This report is generated when the HR admin wants to verify the existing I9 form records.

Verify I9 Form: HR Administrator

Form ID	SUID	HR ID	Employer ID	Location Issued	Date Issued
1	200	100	300	Steele Hall	2010-03-03
2	201	100	301	Steele Hall	2010-12-03
3	202	100	302	Steele Hall	2010-12-04
4	203	100	303	Steele Hall	2010-12-05
5	204	100	304	Steele Hall	2010-12-05
6	205	100	305	Steele Hall	2010-12-05
7	206	100	306	Steele Hall	2010-12-05
8	207	100	307	Steele Hall	2010-12-05
9	208	100	308	Steele Hall	2010-12-05
10	209	100	309	Steele Hall	2010-12-05

Student → Generate Employer Count

This report is generated when a student wishes to calculate the total number of employers he works under.

Total number of Employers

SUID	FirstName	LastName	Total number of Employers
200	Sachin	Tendulkar	1

Student → Check Total Income

This report is generated when a student wishes to calculate his total income.

Total income

SUID	Total Salary in Dollars
200	10

Student → Employment Letter Status:

This report is generated when a student wishes to check his employment letter status.

Employment letter status

SUID	Letter ID	Job Type	Date issued	Salary Paid
200	1	Supervisor	2011-02-02	10

Student → Verify I9 Form status

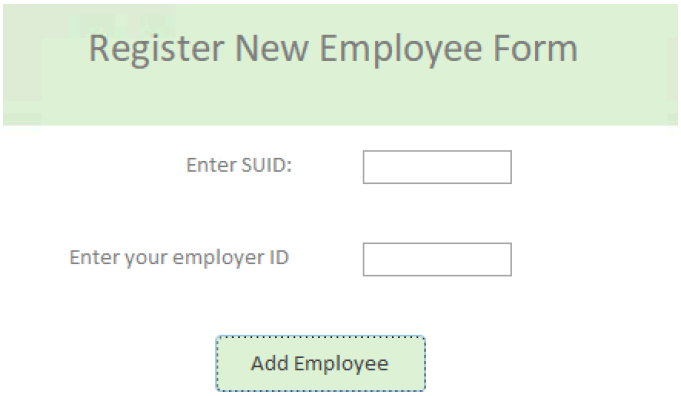
This report is generated when a student wishes to verify his I9 form status.

I9 Student Form Status

Form ID	SUID	First Name	Last Name	HR ID	Employer ID	Location	Date issued
1	200	Sachin	Tendulkar	100	300	Steele Hall	2010-12-02

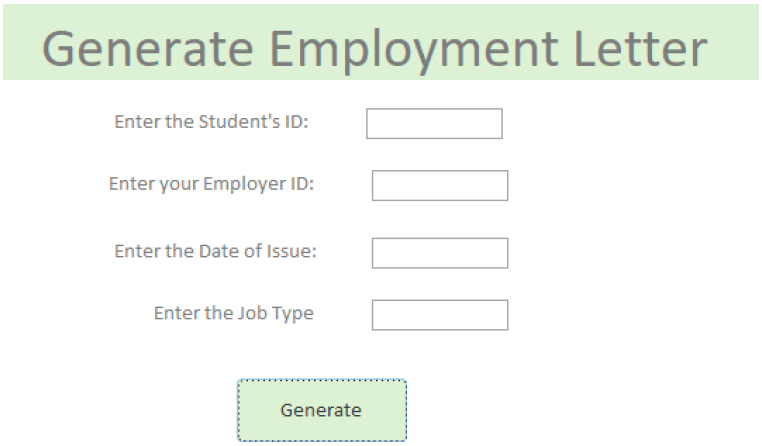
Employer → Add student employee Form

This form allows an employer to register a new student employee.



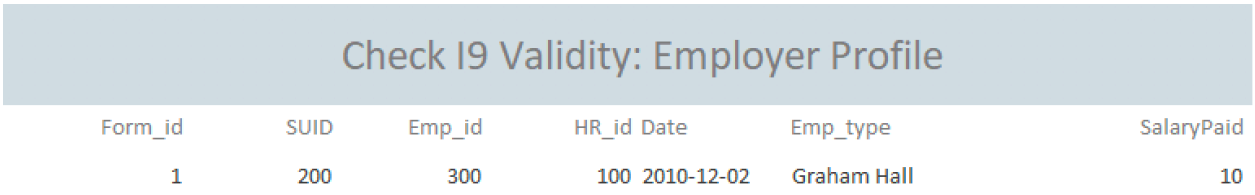
Employer → Generate Employment Letter

This form allows an employer to generate an employment letter for a student employee.



Employer → Check I9 Form validity

This report is generated when an employer wishes to check the I9 form validity of a student employee.



Trigger:

The trigger that has been implemented is used to update the employer count value of a Student employee every time he gains employment.

Logic: Whenever the employer updates his database by adding a student employee to his records, the Student table's employer count attribute automatically gets incremented for that particular student.

Code:

```
CREATE TRIGGER Update_employercount
ON StudentJob_record
FOR INSERT, UPDATE
AS
IF @@ROWCOUNT >= 1
BEGIN
UPDATE Student
SET  employer_count = temp.employer_count
FROM
(SELECT s.SUID ,count(Emp_id) AS 'employer_count'
FROM StudentJob_record sj
inner join Student s
ON s.SUID = sj.SUID
GROUP BY s.SUID) AS temp
WHERE Student.SUID=temp.SUID
END;
```

Before running the trigger:

Student Table:

	SUID	employer_count
1	200	1
2	201	1
3	202	1
4	203	1
5	204	1
6	205	1
7	206	1
8	207	1
9	208	1
10	209	1

StudentJob_record Table:

	App_record_id	SUID	Emp_id
1	1	200	300
2	2	201	301
3	3	202	302
4	4	203	303
5	5	204	304
6	6	205	305
7	7	206	306
8	8	207	307
9	9	208	308
10	10	209	309

After running the trigger:

```
INSERT INTO StudentJob_record VALUES(11,200,302);
INSERT INTO StudentJob_record VALUES(12,200,303);
```

Student Table:

	SUID	employer_count
1	200	3
2	201	1
3	202	1
4	203	1
5	204	1
6	205	1
7	206	1
8	207	1
9	208	1
10	209	1

StudentJob_record Table:

	App_record_id	SUID	Emp_id
1	1	200	300
2	2	201	301
3	3	202	302
4	4	203	303
5	5	204	304
6	6	205	305
7	7	206	306
8	8	207	307
9	9	208	308
10	10	209	309
11	11	200	302
12	12	200	303