CPSC471 - PAYROLL SYSTEM FINAL REPORT

PREPARED FOR

Dr. Jalal Kawash

Kashfia Sailunaz

PREPARED BY

Abhishek Balasubramanian
Omar Qureshi
Hamzah Umar

1. Abstract

Accounting for payroll is a major issue for many companies. Employers need a way to keep track of employee salaries, benefits and deductions to ensure smooth payroll operations. The project aims to tackle these issues by designing a modest web interactable database. Using a local host connection for exercise purposes, employers can either register or login, and employees can simply login into the system. As an employer, there is greater website functionality due to options such as adding employee, adjusting their details and performing certain functions on tables via SQL queries. Employees when logged in can simply view their own details pulled from the appropriate tables and then displayed the appropriate rows given their SIN and employee ID.

Overall, the system was designed with the original project proposal specification in mind, albeit with a few minor changes.

2. Introduction

A) The Problem

Businesses and companies have been a significant part of human societies for centuries now. One direct consequence of managing a company is having to pay your employees a salary. Therefore, it is important for a company to keep a record of all their employees, and all information related to their salary. Traditionally, this has been done manually by accountants using pen and paper. In recent times however, there has been a shift towards using automated systems. Automated, computer-based systems are typically preferred because they save time and they reduce the risk of calculation errors. Thus arises the problem: **How can a company easily and conveniently set up an automated computer-based system to manage its payroll?**

This problem is interesting because it has rather significant ramifications. If solved incorrectly, a company can leave itself open to litigation if it ends up having inaccurate information in its payroll.

The problem usually arises in one of the 2 scenarios

- An existing company which already has a payroll that is manually recorded and wants to transition to a computer-based database
    - A recent survey by Clutch showed that 25% of small businesses still record their finances manually on paper. As we continue into the future, it’s plausible that many of these companies may decide at one point to upgrade to a computer-based system.
- A newly formed company which wants to set up a computer-based payroll database from scratch
    - Ideally, a new company would want to set off on the right foot and simply jump straight to an automated database.

In practice, the problem that we solved had already been solved. There are programs which give managers the ability to create and manage a payroll on a computer. The most common solution offered by software development companies is to create PC applications which can be installed and operated by a business manager/owner. There are also some web applications which offer similar functionality, but they are far less common.

One system which is similar to the one we created is called Intuit Payroll. This is a system which can automatically generate paychecks as well as calculate tax deductions and other details relating to the paycheck. Intuit Payroll also has a feature that allows managers to chat online with ‘payroll specialists’ in case of any difficulties.

One thing we noticed when we were researching current solutions to the problem is that they were almost exclusively designed for the management side. In other words, most automated payroll systems are designed to only be interacted with by company managers/owners. To improve upon this, we set out with the goal of creating a portable system which can also be used by employees (albeit with fewer privileges obviously). We wanted our system to be accessible to employers and employees alike, but we wanted employees to only have “read-only” access to the system.

B) The Solution

We produced a centralized automated payroll management system. All the underlying data of the website was saved in a database as required, and we developed a companion website which serves as the front-end for the system and is used to access, view and modify the data in the database.

The automated nature of the database means that it is less prone to human recording errors and thus achieves better accuracy. It also tracks and limits the people who access the database to make changes, which increases security and holds managers who make changes accountable to what they did. This is a marked improvement from the ledger system of managing payroll, which can be modified by someone to sabotage the company. As 8% of companies make manual entry mistakes in the payroll system and 48% of companies do not even notice such a mistake until an employee contacts management about it, this system not only offers valuable measures to prevent the data integrity of the payroll system from being compromised, but also improves the maintainability of the company payroll. The project can be expanded to have as many employees as required at no cost design-wise.

The features of this system include:

- *Calculating salary paid after tax deductions:* This feature is used to calculate the effective amount of money sent to employees and keeps track of the amount of money to be sent to the Government as part of taxes. It is also useful for allocating money to other programs such as Union Dues, Employment Insurance etc. This is done by using the base salary and performing operations on it to separate the taxes and net salary.
- *Adding and removing employees from the payroll system*: This feature is used when new employees join a company and when existing employees terminate their services. This feature is implemented with a query to add or remove an entry and is done such that all other tables that have the employee’s key remove that particular entry as well to keep the database in a consistent state.
- *Increasing and decreasing the salary of an employee; editing features of employees and payments*: This feature is used to offer employees a pay raise or offer bonuses.

This feature is implemented with a query that updates the base salary attribute of an employee.

- *Searching for employees based on their data:* A feature that allows searching for employees with certain markers. This allows the company to narrow down on who is paid the highest or whether a certain subset of employees is receiving adequate benefits. This feature uses a search query that can accept existing attributes of employees to search and display results.
- *Maintaining a login system*: This feature allows managers or administrators of a company to make their own account to manage the payroll system. The username and password are stored in the database itself and entries to log in are checked against that table.

3. Users of System

There are two types of users that can log into the payroll system: Employers and Employees.

Employers

An employer has the ability to register their company on the registration page. An employer registers by entering:

- A company name
- Their own personal details

Upon doing so, they are provided with automatically generated login details, which they can use to log in to the website.

Upon logging in, an employer has unrestricted access to all parts of the database and can edit most details.

Managing Users/Employees

In the system, an employer is responsible for managing their employees. This includes adding new employees, deleting employees, or editing the details of existing employees. To add an employee, an employer must enter:

- The employee's personal details
    - SIN, Name, Phone Number, Number of Minor Dependents

Whenever an employer adds an employee, the system automatically generates login details for that employee, which they can use to log into the website themselves.

Managing Salaries

An employer can also add a salary for that employee. Each salary has:

- An annual amount
- A frequency (how often this salary is to be paid; monthly, weekly etc.)

The employer can edit this salary whenever needed.

Managing Bonuses

An employer has the option of giving a monetary bonus to any employee that they have added to the system. When giving a bonus, an employer provides

- The monetary value of the bonus
- The reason for the bonus being awarded

Like a salary, a bonus can be edited whenever required.

Managing Benefits & Deductions

The system takes it upon itself to calculate benefits and deductions automatically based on an employee's details and their salary. However, we give an employer the option to manually edit these figures if they ever need to. Specifically, a benefit contains

- CCB benefits based on the number of minor dependents an employee has
- Healthcare benefits

A deduction contains

- Income tax and employment insurance deductions based on an employee's salary
- Union dues

Viewing Reports

Based on all the information entered in the sections described above, the system will automatically generate reports that calculates:

- the net annual salary for each employee
- the net salary each employee is due at the end of the current month (including any bonuses received within the last month)
- the total annual payroll for the company
- the average net annual salary of all the company's employees
- a breakdown of average net annual salary grouped by sex

Employees

On the contrary, employees have much less access to the data. They are unable to edit any data, and can only view information that relates to them personally. Lastly, an employee can’t register an account for themselves. Rather, they have to wait for their employer to enter their details into the system and then provide them with login details

Viewing Employee Information

Upon logging in, an employee can view all the information that pertains to them. Specifically, they can view:

- Their personal information.
- Their net salary (after benefits and deductions)
- The constituent parts of their net salary
    - Their base salary
    - The specific amount for each benefit
    - The specific amount for each deduction

4. Enhanced Entity Relationship Diagram (EERD)

We underwent an iterative process when it came to our EERD. Some of the changes we made from our previous submissions included omitting some superfluous attributes such as currency. We assumed it was a Canadian payroll system and supported only one currency for the sake of simplicity. Other changes included using a merging the super/subclasses via the algorithm (step 8c). This made the relational model a little simpler and as such the SQL as well due to fewer tables. Our final EERD can be viewed below;

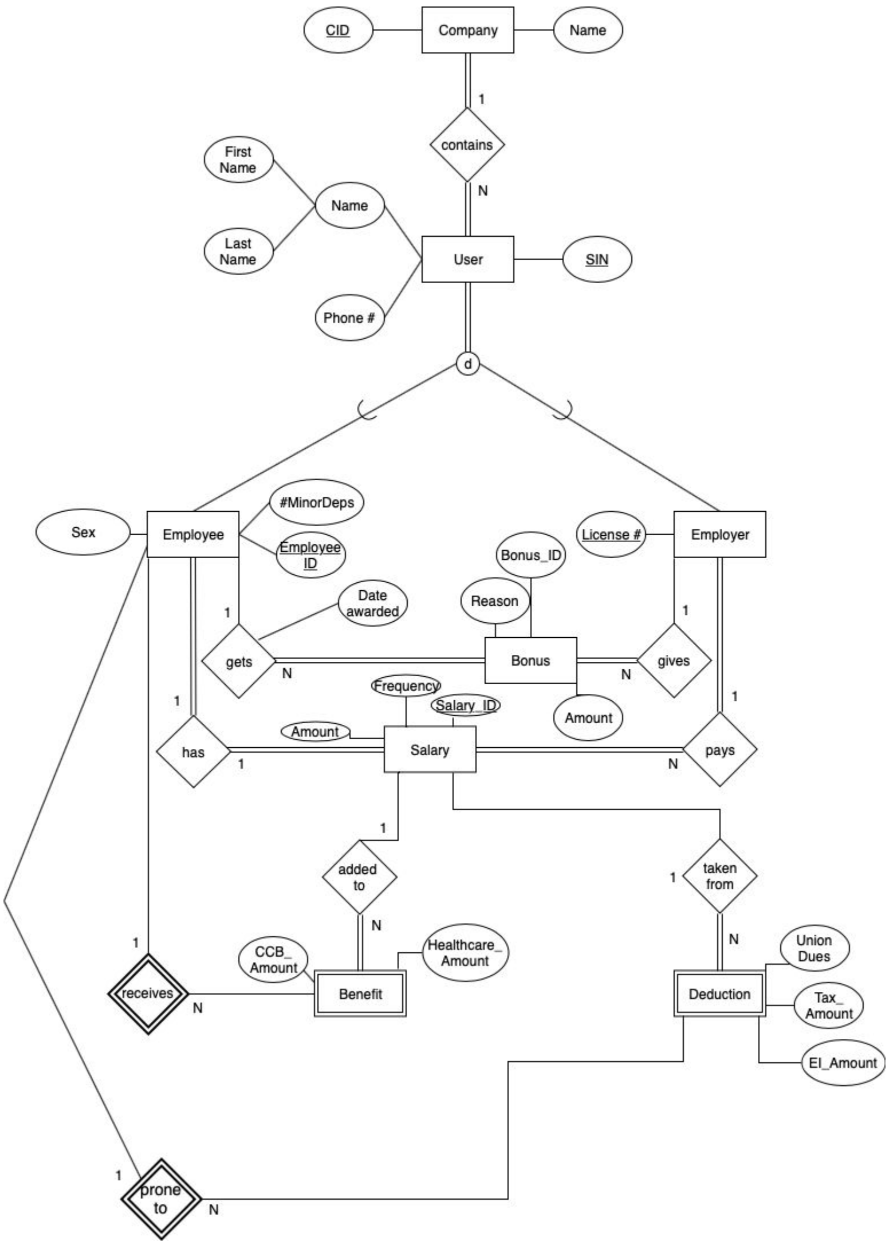
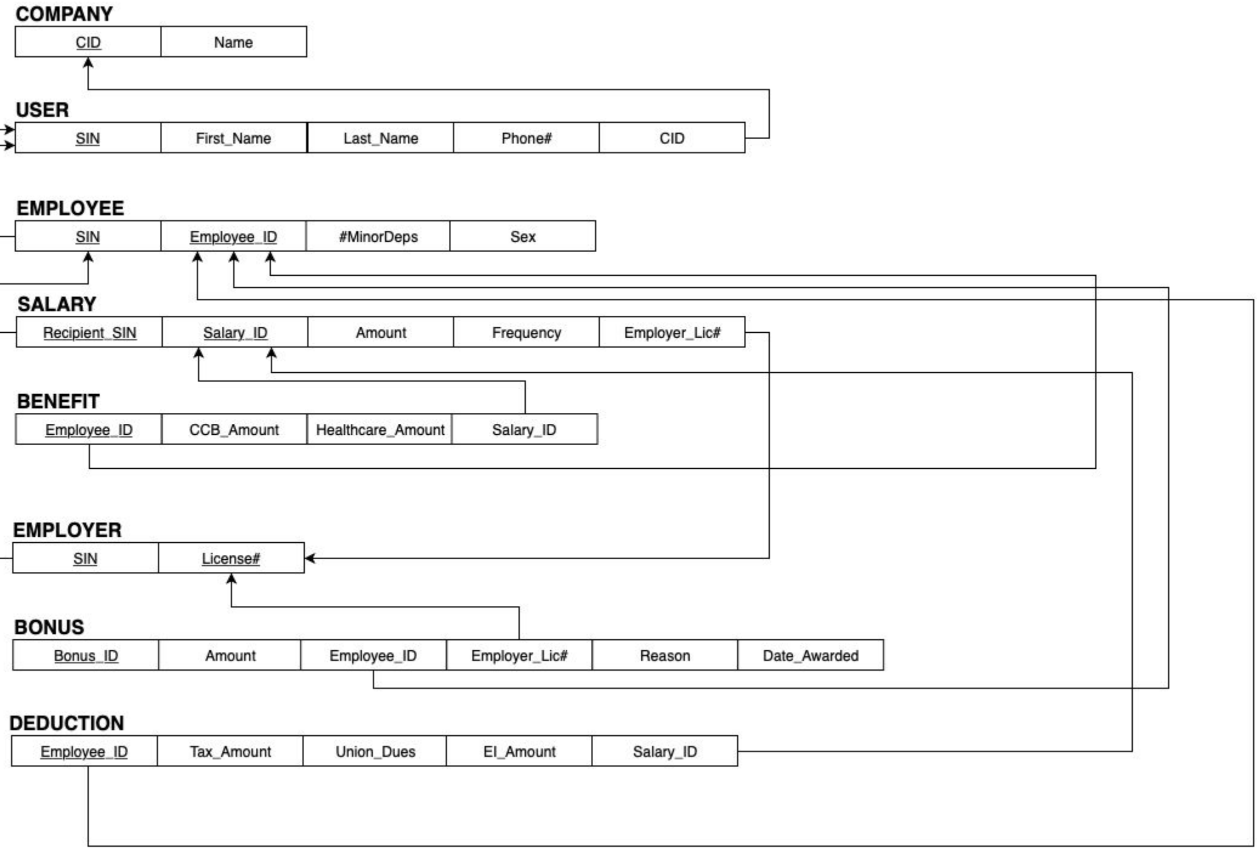


Figure1: EER diagram

5. Relational Model (RM)

As with our EERD, the RM diagram correspondingly went through iterative changes as well. The most notable ones like mentioned were the merging of the sub/superclasses. In addition, we decided to scale back on the functionality of the system due to time constraints. As such, the log functionality was dropped since it did not seem pertinent to how the system behaves as a whole. Our most recent version of the relational model can be viewed below;



COMPANY

CID	Name

USER

SIN	First_Name	Last_Name	Phone#	CID

EMPLOYEE

SIN	Employee_ID	#MinorDeps	Sex

SALARY

Recipient_SIN	Salary_ID	Amount	Frequency	Employer_Lic#

BENEFIT

Employee_ID	CCB_Amount	Healthcare_Amount	Salary_ID

EMPLOYER

SIN	License#

BONUS

Bonus_ID	Amount	Employee_ID	Employer_Lic#	Reason	Date_Awarded

DEDUCTION

Employee_ID	Tax_Amount	Union_Dues	EI_Amount	Salary_ID

Figure 2: Relational Model diagram

6. HIPO & DF Diagrams

The major changes reflected on our HIPO and DF diagrams compared to our previous submission was the omission of the log feature. This simplified the design and allowed us to polish the system better. Our most recent HIPO and DF diagrams can be viewed below;

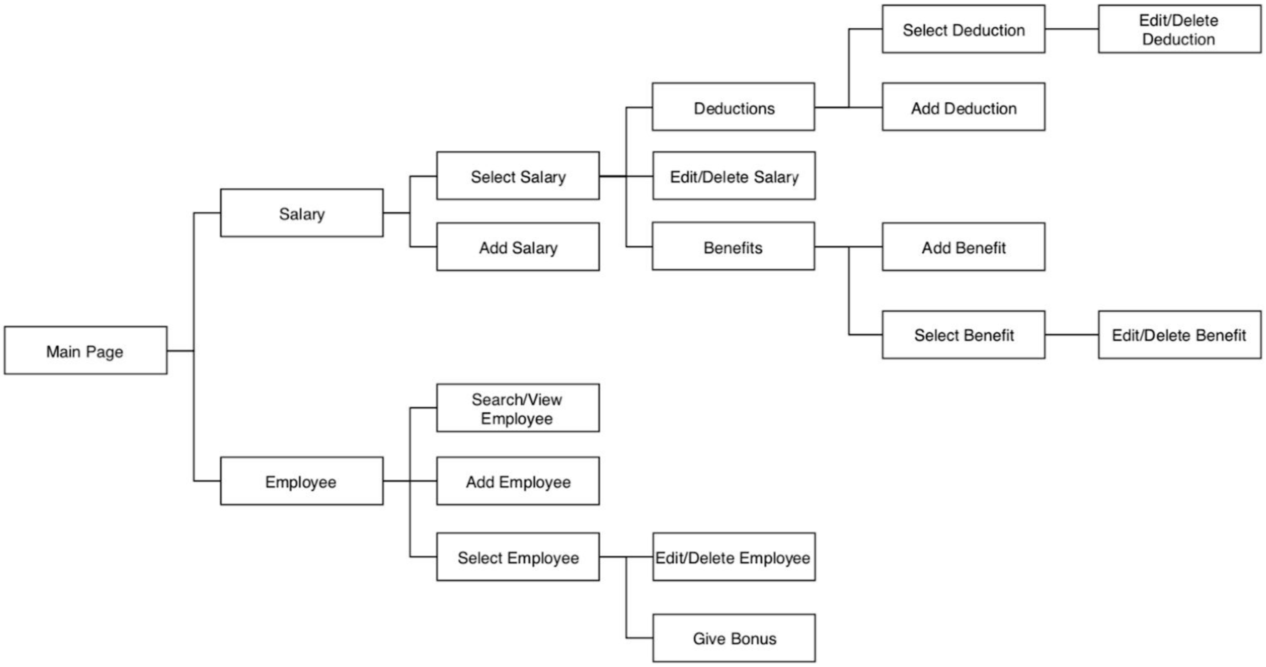


Figure 3: HIPO diagram

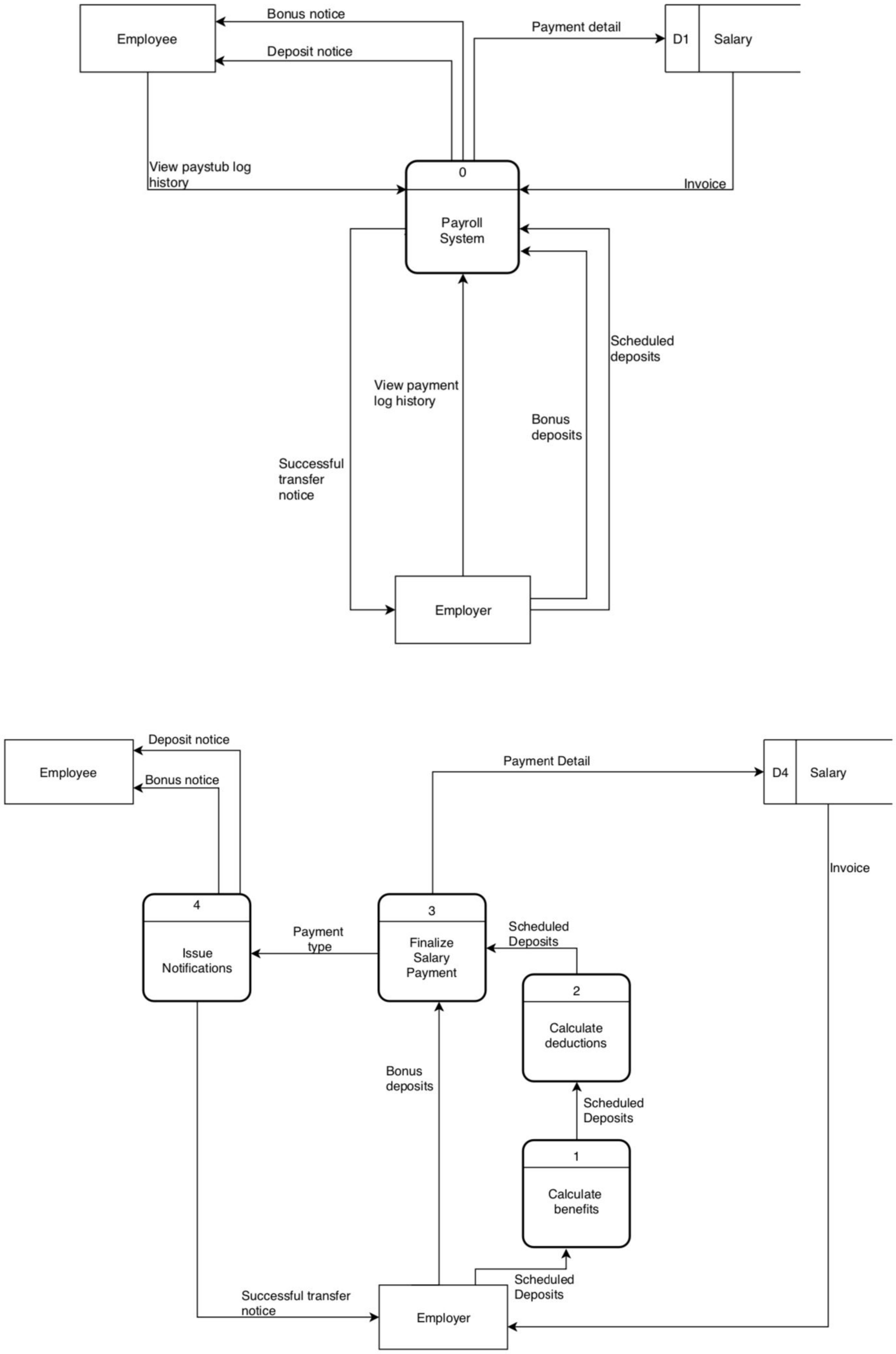


Figure 4: Data flow diagrams (context diagram above, level 0 diagram below)

7. Database Management System (DBMS)

The implementation began by first installing MySQL server along with MySQL workbench. Once the MySQL server was running, the instance could be picked up by workbench to begin creating tables with appropriate constraints. The workbench application provided a decent GUI environment for creating the Payroll schema and creating respective tables along with populating them with their attributes. We could also set constraints and enable how entries should behave on update, insert, delete.

With the database set up, we created a server using NodeJs, and connected the server to the database. From there, we used Socket.io to establish a connection between the client-side and the server-side.

The following is a list of SQL statements used:

Register a company

- INSERT

  INTO company (CID, Name)

  VALUES (?, ?), [CID, aCompanyName]

Register an employer

- INSERT

  INTO user (SIN, First_Name, Last_Name, PhoneNum, CID, password)

  VALUES (?, ?, ?, ?, ?, ?), [aSIN, aFirstName, aLastName, aPhoneNum, CID, aPassword]
- INSERT

  INTO employer (SIN)

  VALUES (?), [aSIN]

Login

- SELECT loginID, password

  FROM user

  WHERE loginID = (?) AND password = (?), [data.loginID, data.password]

List companies

- SELECT * FROM company

Edit a company

- UPDATE company

  SET Name = (?)

  WHERE CID = (?), [aNew_Name, aCID]

List users

- SELECT * FROM user

Edit user

- UPDATE user

  SET First_Name = (?), Last_Name = (?), PhoneNum = (?)

  WHERE SIN = (?), [aFirstName, aLastName, aNew_Phone_Num, aSIN]

List employees

- SELECT SIN, Employee_ID, NumMinorDeps, Sex

  FROM employee

  ORDER BY SIN

Add employee

- INSERT

  INTO user (SIN, First_Name, Last_Name, PhoneNum, CID, password)

  VALUES (?, ?, ?, ?, ?, ?), [aSIN, aFirstName, aLastName, aPhoneNum, aCID, aPassword

- INSERT

  INTO employee (SIN, NumMinorDeps, Sex)

  VALUES (?, ?, ?), [aSIN, aNumberDependents, aSex]

Edit employee

- UPDATE employee

  SET NumMinorDeps = (?), Sex = (?)

  WHERE SIN = (?), [aNew_Number_Minor_Deps, aNew_Sex, aSIN]

Delete employee

- DELETE from user

  WHERE SIN = (?), [aSIN]

List salaries

- SELECT * FROM salary

Add salary

- INSERT

  INTO salary (Recipient_SIN, Employer_Lic#, Amount, Frequency)

  VALUES (?, ?, ?, ?), [aSIN, anEmployer, anAmount, aFrequency]

- INSERT

  INTO deduction (Employee_SIN, Salary_ID, Tax_Amount, EI_Amount, Union_Dues)

  VALUES (?, ?, ?, ?, ?), [aSIN, salary_ID, anAmount * tax_percentage, anAmount * ei_percentage, 20]

- INSERT

  INTO benefit (Employee_SIN, Salary_ID, CCB_Amount, Healthcare_Amount)

  VALUES (?, ?, ?, ?), [aSIN, salary_ID, baseCCB * numMinorDeps, 2000]

Edit salary

- UPDATE salary

  SET Amount = (?), Frequency = (?)

  WHERE Salary_ID = (?), [aNew_Amount, aNew_Frequency, aSalaryID]

List bonuses

- SELECT * FROM bonus

Add bonus

- INSERT

  INTO bonus (Employee_ID, Employer_Lic#, Reason, Date_Awarded, Amount)

  VALUES (?, ?, ?, ?, ?), [anEmployee, anEmployer, aReason, new Date() , anAmount]

Edit bonus

- UPDATE bonus

  SET Amount = (?), Reason = (?)

  WHERE Bonus_ID = (?), [aNew_Amount, aNew_Reason, aBonus]

List benefits

- SELECT Salary_ID, Employee_SIN, CCB_Amount, Healthcare_Amount

  FROM benefit

  ORDER BY Salary_ID

Edit benefit

- UPDATE benefit

  SET CCB_Amount = (?), Healthcare_Amount = (?)

  WHERE Salary_ID = (?), [aNew_CCBAmount, aNew_HealthcareAmount,aSalaryID]

List deductions

- SELECT Salary_ID, Employee_SIN, Tax_Amount, EI_Amount, Union_Dues

  FROM deduction

  ORDER BY Salary_ID

Edit deduction

- UPDATE deduction

  SET Tax_Amount = (?), EI_Amount = (?), Union_Dues = (?)

  WHERE Salary_ID = (?), [aNew_TaxAmount, aNew_EIAmount, aNew_UnionAmount, aSalary_ID]

List net salaries

- SELECT user.SIN, employee.Employee_ID, user.First_name, user.Last_Name, (salary.Amount + benefit.CCB_Amount + benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues) AS Net_Salary

  FROM user, employee, salary, benefit, deduction

  WHERE user.SIN = employee.SIN AND salary.Recipient_SIN = employee.SIN AND benefit.Salary_ID = salary.Salary_ID AND deduction.Salary_ID = salary.Salary_ID

List salary stats

- SELECT SUM(salary.Amount + benefit.CCB_Amount + benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues) AS Total_Annual_Payroll, AVG(salary.Amount + benefit.CCB_Amount + benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues) AS Average_Annual_Salary

  FROM user, employee, salary, benefit, deduction

  WHERE user.SIN = employee.SIN AND salary.Recipient_SIN = employee.SIN AND benefit.Salary_ID = salary.Salary_ID AND deduction.Salary_ID = salary.Salary_ID

List average by sex

- SELECT employee.Sex, AVG(salary.Amount + benefit.CCB_Amount, benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues) AS Average_Annual_Salary

  FROM user, employee, salary, benefit, deduction

  WHERE user.SIN = employee.SIN AND salary.Recipient_SIN = employee.SIN AND benefit.Salary_ID = salary.Salary_ID AND deduction.Salary_ID = salary.Salary_ID

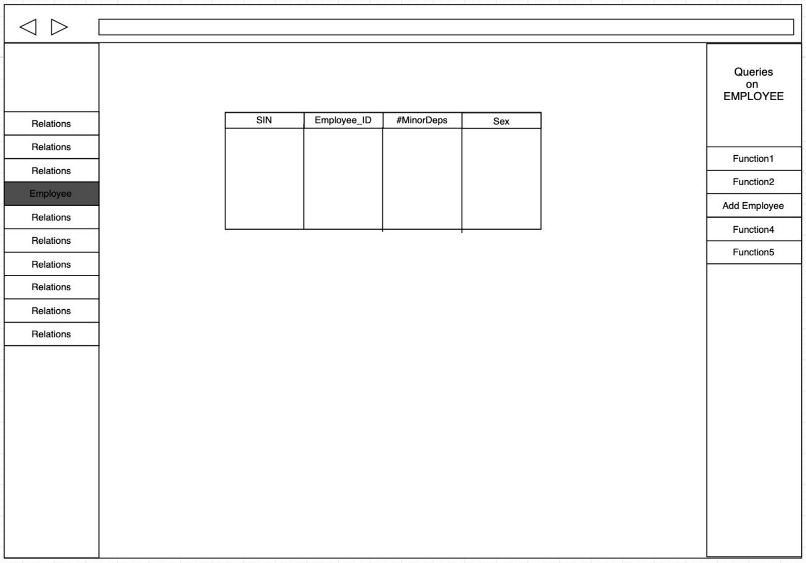
  GROUP BY employee.Sex

Due this month (most complex query)

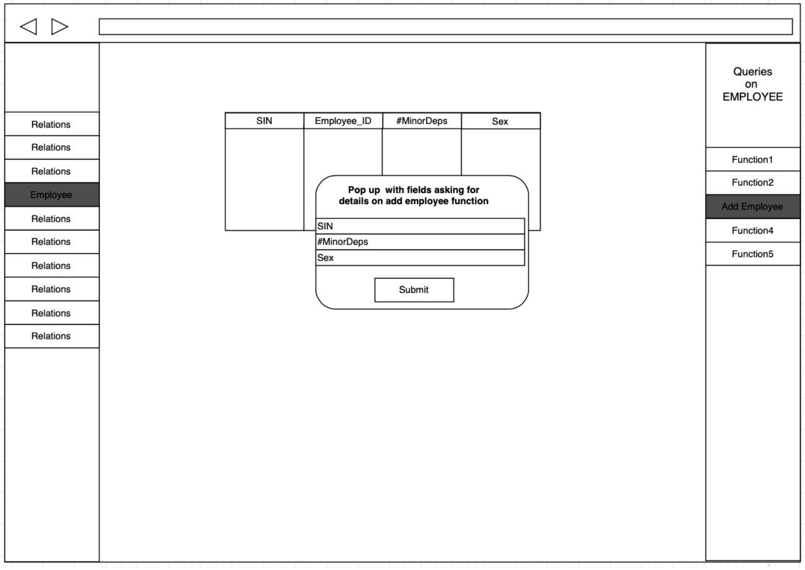
-
  ```
  SELECT user.SIN, employee.Employee_ID, user.First_Name, user.Last_Name, (((salary.Amount + benefit.CCB_Amount + benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues)/(?)) + SUM(bonus.Amount)) AS Due_This_Month

  FROM user, employee, salary, benefit, deduction, bonus

  WHERE user.SIN = employee.SIN AND user.SIN = salary.Recipient_SIN AND user.SIN = benefit.Employee_SIN AND user.SIN = deduction.Employee_SIN AND salary.frequency = (?) AND bonus.Employee_ID = employee.Employee_ID AND MONTH(bonus.Date_Awarded) = MONTH(curdate())

  GROUP BY user.SIN

  UNION

  SELECT user.SIN, employee.Employee_ID, user.First_Name, user.Last_Name, ((salary.Amount + benefit.CCB_Amount + benefit.Healthcare_Amount - deduction.Tax_Amount - deduction.EI_Amount - deduction.Union_Dues)/(?)) AS Due_This_Month

  FROM user, employee, salary, benefit, deduction, bonus

  WHERE user.SIN = employee.SIN AND user.SIN = salary.Recipient_SIN AND user.SIN = benefit.Employee_SIN AND user.SIN = deduction.Employee_SIN AND salary.frequency = (?) AND employee.Employee_ID NOT IN (SELECT bonus.Employee_ID FROM bonus)
  ```

8. User Manual

UI Mockups



SIN	Employee_ID	#MinorDeps	Sex



SIN	Employee_ID	#MinorDeps	Sex

Pop up with fields asking for details on add employee function

SIN

#MinorDeps

Sex

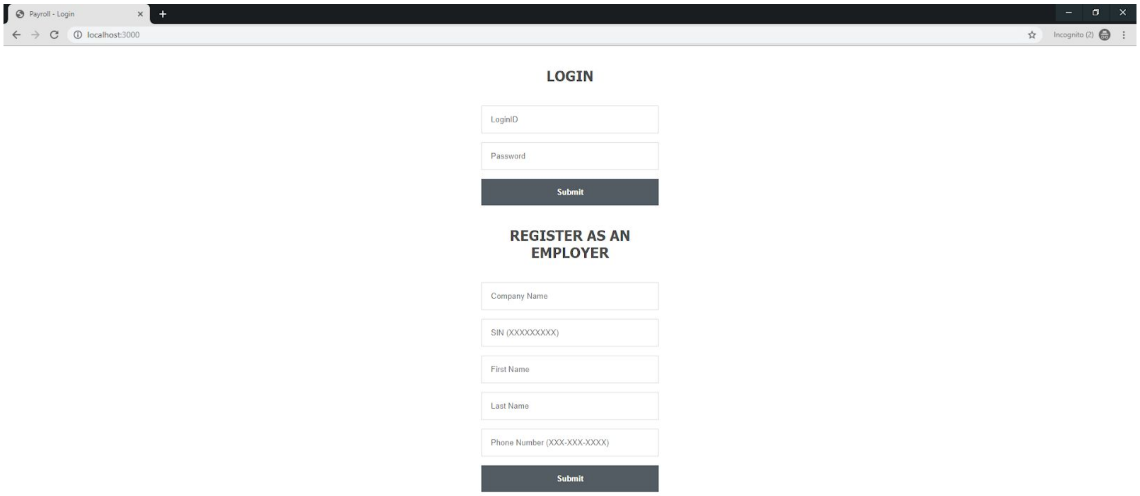
General Layout

In a nutshell, there are three major components to the main UI of the website's subpages;

- the left relations sidebar,
- the current relation table view in the middle and,
- the right queries sidebar.

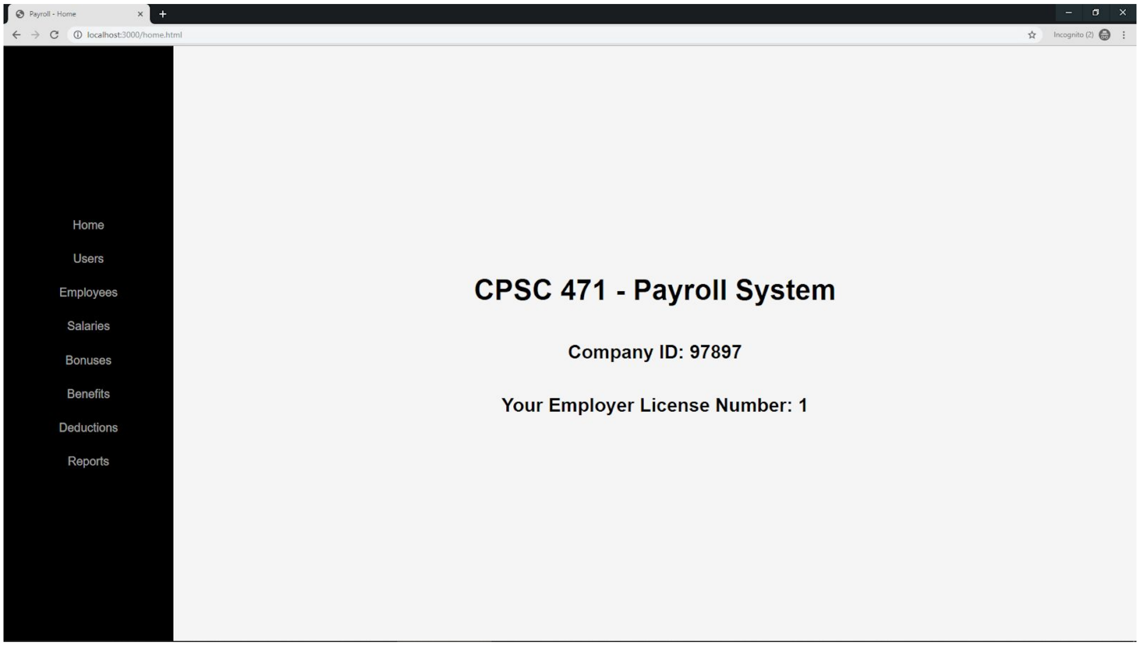
Selecting a relation from the left will bring up its table view along with the appropriate queries for that relation. Clicking on a query will then bring up a pop up menu that'll take certain fields to execute the query and modify the database which would possibly result in a new view of the relation table in the middle. This is the main foundation of how users will interact with the system and further details on each specific page will now be explored.

Login/Registration Page



When a user first loads up the site, they are greeted with a login page. If the user has an account already, they can simply enter their details and log in, Otherwise, an employer can register an account and then log in. The website is designed in such a way that an employee never register their own account, rather it is automatically generated when an employer adds them. This will be described further later.

Home Page



CPSC 471 - Payroll System

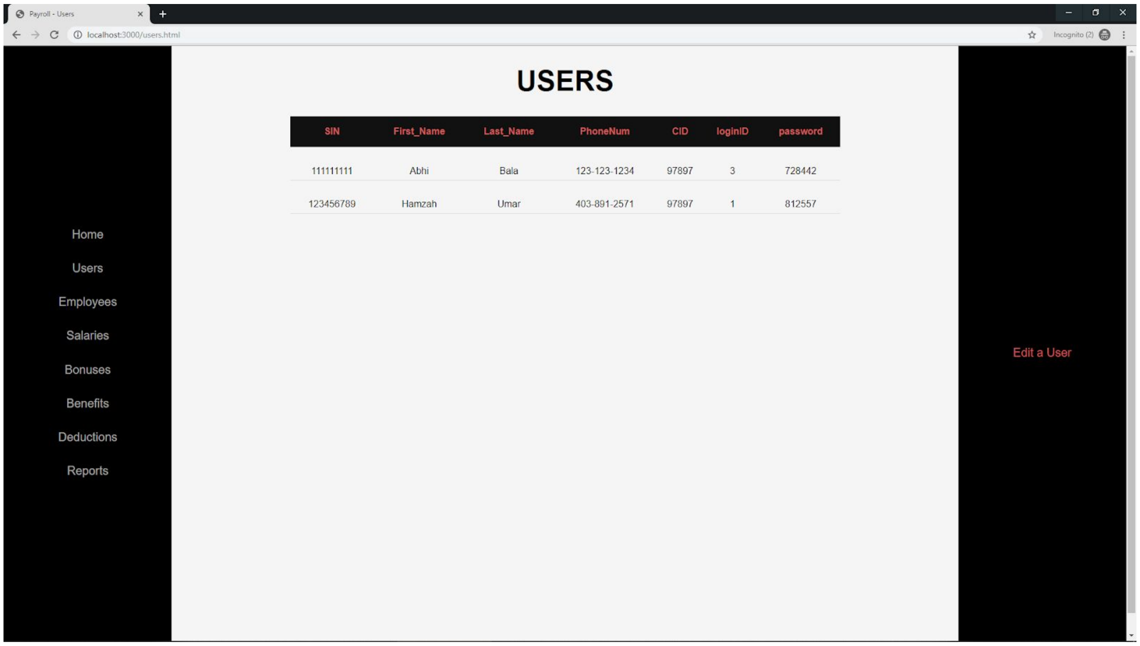
Company ID: 97897

Your Employer License Number: 1

You can click on any option on the left sidebar to begin:

- To view and edit user personal details, click users
- To view, add, edit or delete employees, click employees
- To view, assign or edit salaries, click salaries
- To view, give or edit bonuses, click bonuses
- To view or edit benefits, click benefits
- To view or edit deductions, click deductions
- To view automatically generated reports, click reports

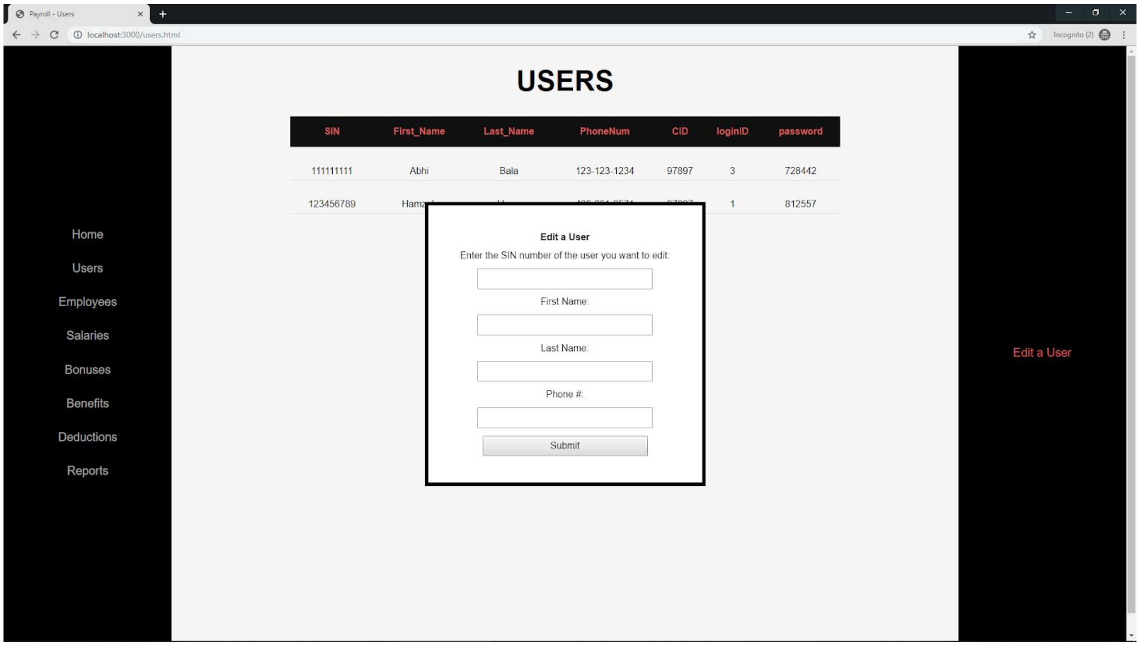
Users Page



USERS

SIN	First_Name	Last_Name	PhoneNum	CID	loginID	password
111111111	Abhi	Bala	123-123-1234	97897	3	728442
123456789	Hamzah	Umar	403-891-2571	97897	1	812557

Use this page to edit the personal information of users.



USERS

SIN	First_Name	Last_Name	PhoneNum	CID	loginID	password
111111111	Abhi	Bala	123-123-1234	97897	3	728442
123456789	Hamz	[illegible]	[illegible]	[illegible]	1	812557

Edit a User

Enter the SIN number of the user you want to edit:

First Name:

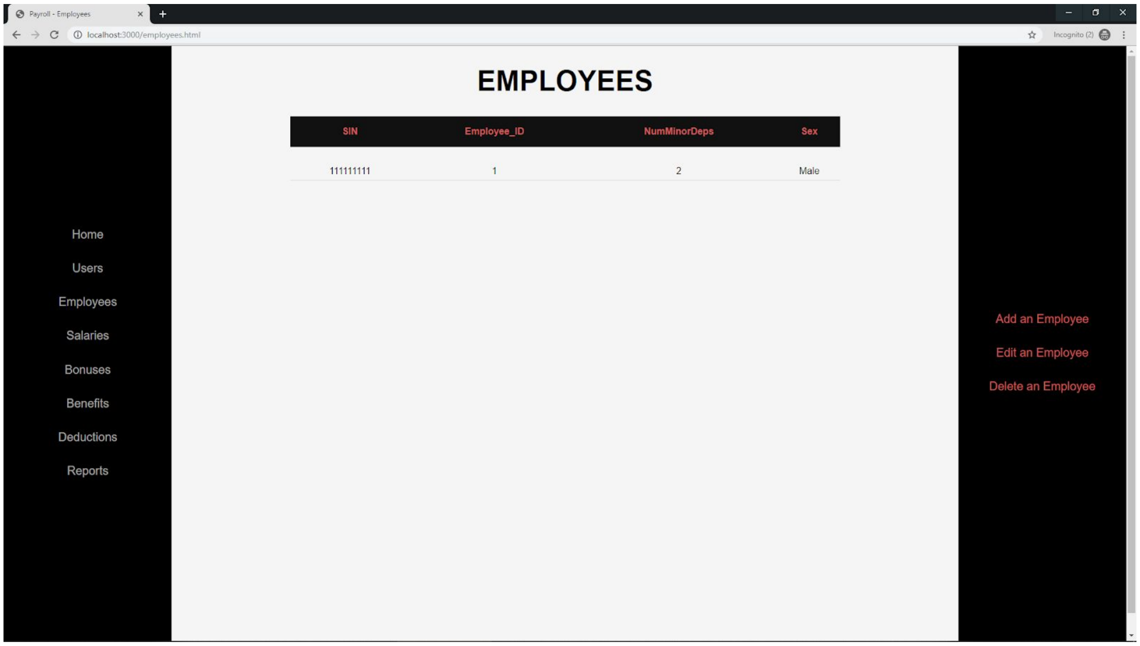
Last Name:

Phone #:

Submit

Edit a user

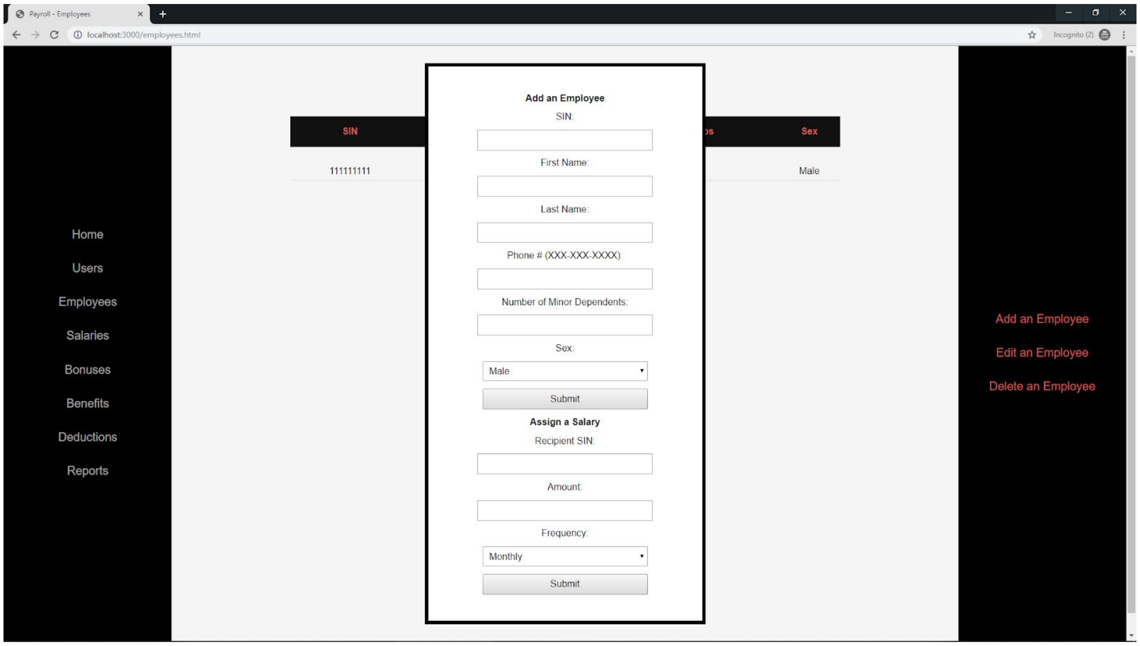
Employees Page



EMPLOYEES

SIN	Employee_ID	NumMinorDeps	Sex
111111111	1	2	Male

Use this page to add/delete employees, or to edit employee-specific information



Add an Employee

SIN:

First Name:

Last Name:

Phone # (XXX-XXX-XXXX):

Number of Minor Dependents:

Sex: Male

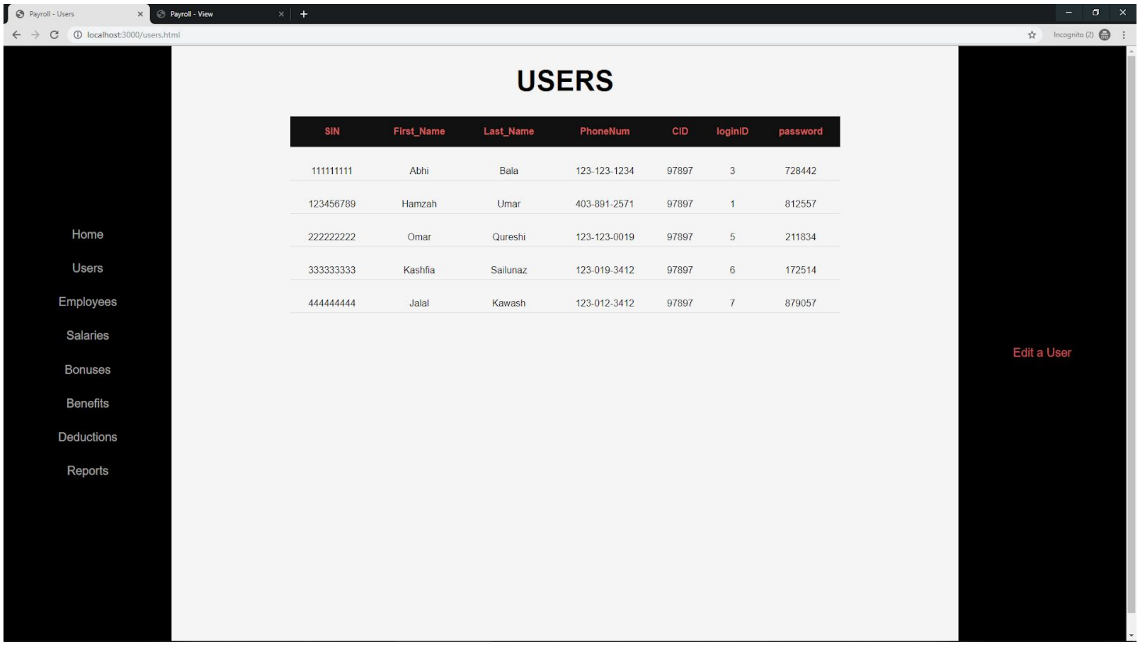
Assign a Salary

Recipient SIN:

Amount:

Frequency: Monthly

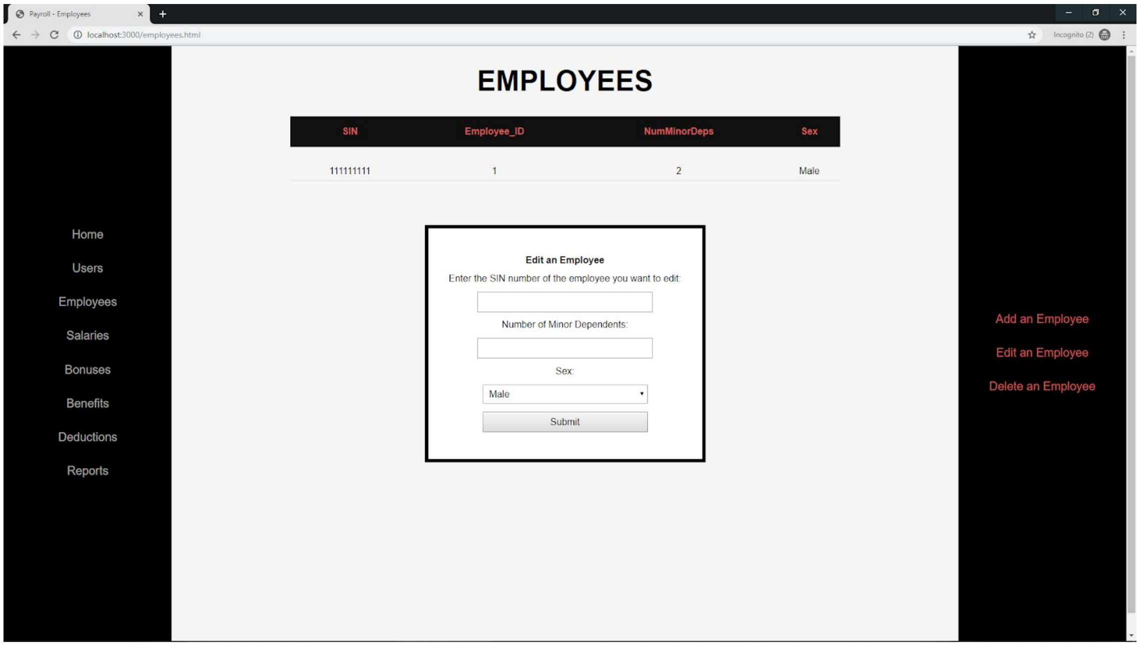
Add an employee. You can also assign a salary when you add an employee.



USERS

SIN	First_Name	Last_Name	PhoneNum	CID	loginID	password
111111111	Abhi	Bala	123-123-1234	97897	3	728442
123456789	Hamzah	Umar	403-891-2571	97897	1	812557
222222222	Omar	Qureshi	123-123-0019	97897	5	211834
333333333	Kashfia	Sailunaz	123-019-3412	97897	6	172514
444444444	Jalal	Kawash	123-012-3412	97897	7	879057

Once you add an employee, the website automatically generates a loginID and a password for that employee. An employer would give this information to their employees, which they can then use to log into the website to view their own information.



EMPLOYEES

SIN	Employee_ID	NumMinorDeps	Sex
111111111	1	2	Male

Edit an Employee

Enter the SIN number of the employee you want to edit:

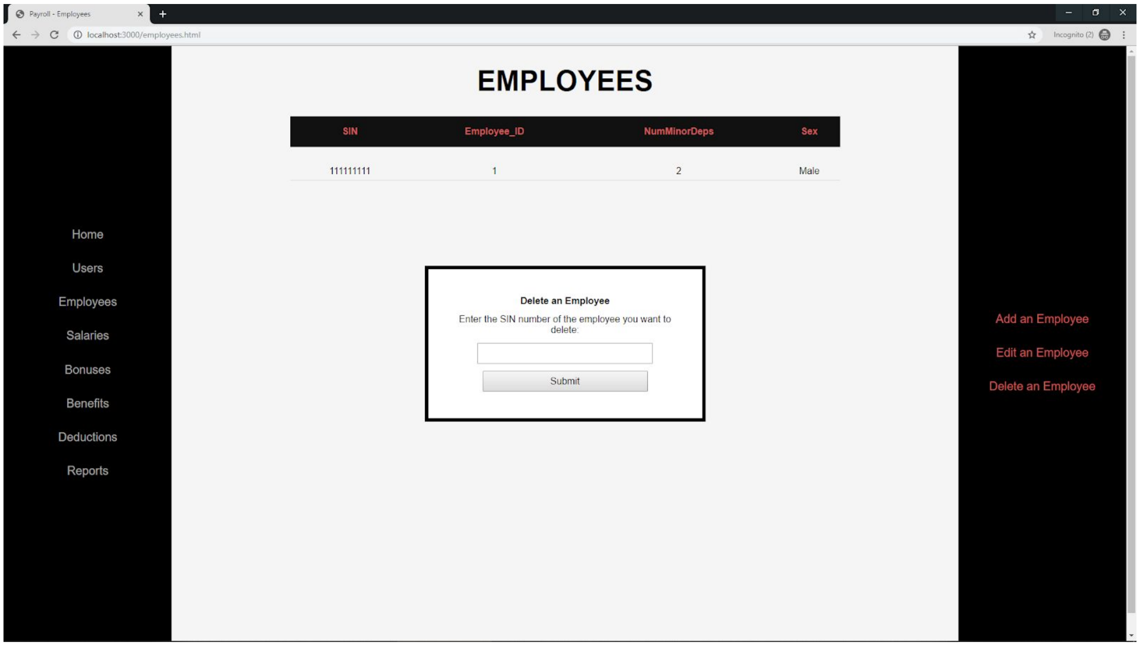
Number of Minor Dependents:

Sex:

Male

Submit

Edit an employee



EMPLOYEES

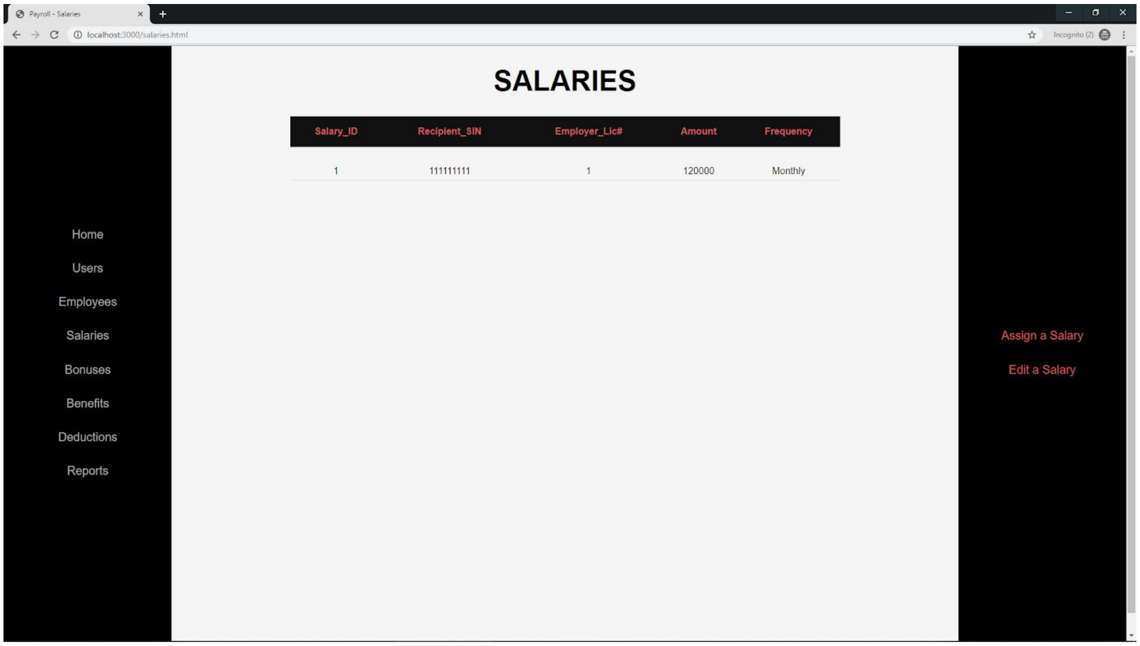
SIN	Employee_ID	NumMinorDeps	Sex
111111111	1	2	Male

Delete an Employee

Enter the SIN number of the employee you want to delete:

Delete an employee

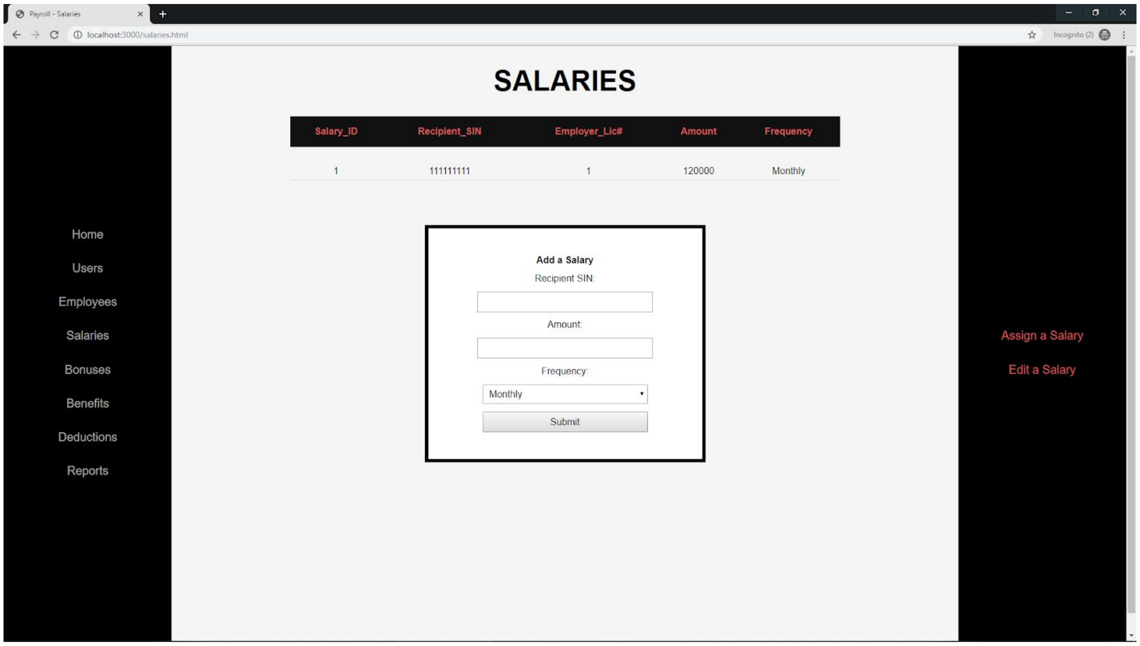
Salaries Page



SALARIES

Salary_ID	Recipient_SIN	Employer_Lic#	Amount	Frequency
1	111111111	1	120000	Monthly

Use this page to assign salaries to employees, or to edit the information of existing salaries



SALARIES

Salary_ID	Recipient_SIN	Employer_Lic#	Amount	Frequency
1	111111111	1	120000	Monthly

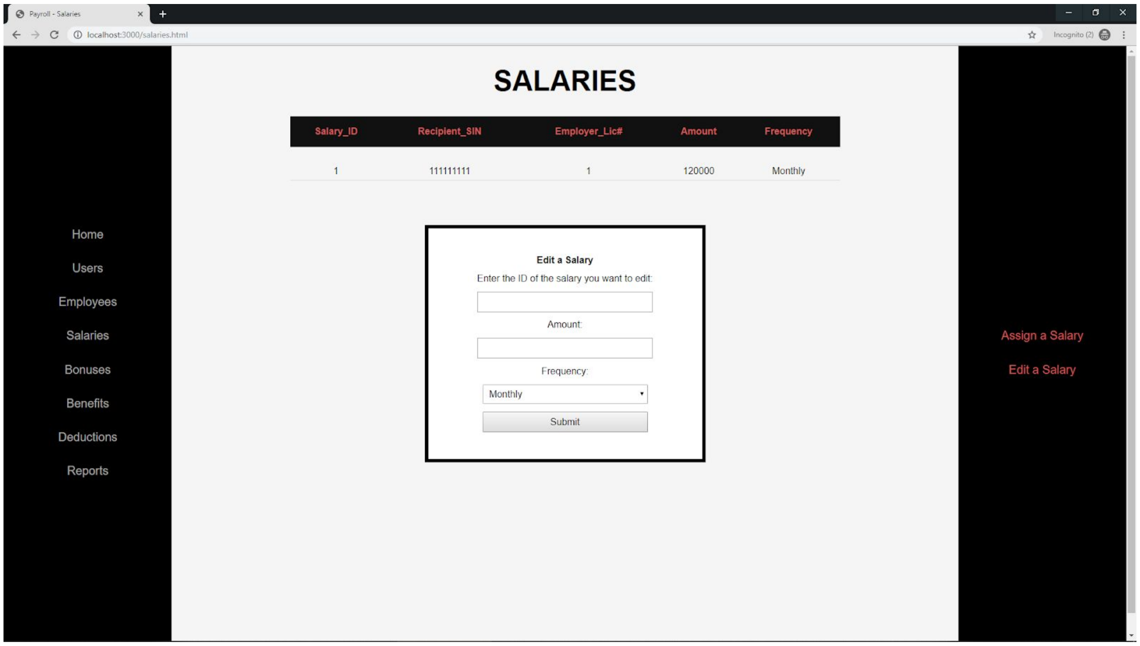
Add a Salary

Recipient SIN:

Amount:

Frequency: Monthly

Assign a salary



SALARIES

Salary_ID	Recipient_SIN	Employer_Lic#	Amount	Frequency
1	111111111	1	120000	Monthly

Edit a Salary

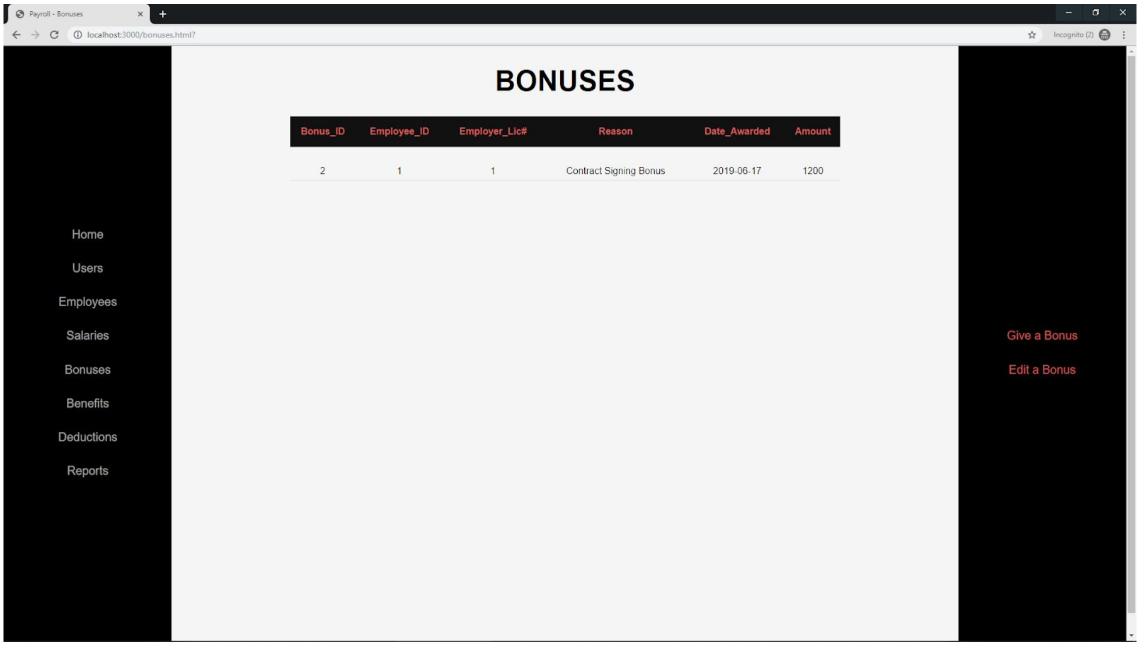
Enter the ID of the salary you want to edit:

Amount:

Frequency: Monthly

Edit a salary

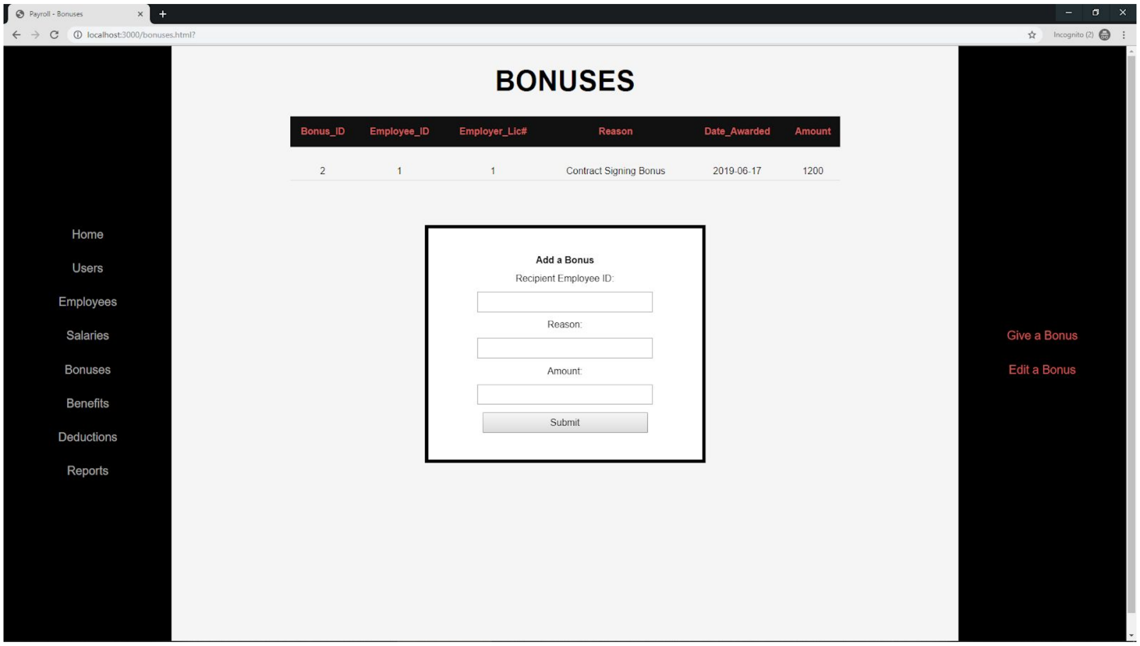
Bonuses Page



BONUSES

Bonus_ID	Employee_ID	Employer_Lic#	Reason	Date_Awarded	Amount
2	1	1	Contract Signing Bonus	2019-06-17	1200

Use this page to give bonuses to employees, or to edit the information of existing bonuses.



BONUSES

Bonus_ID	Employee_ID	Employer_Lic#	Reason	Date_Awarded	Amount
2	1	1	Contract Signing Bonus	2019-06-17	1200

Add a Bonus

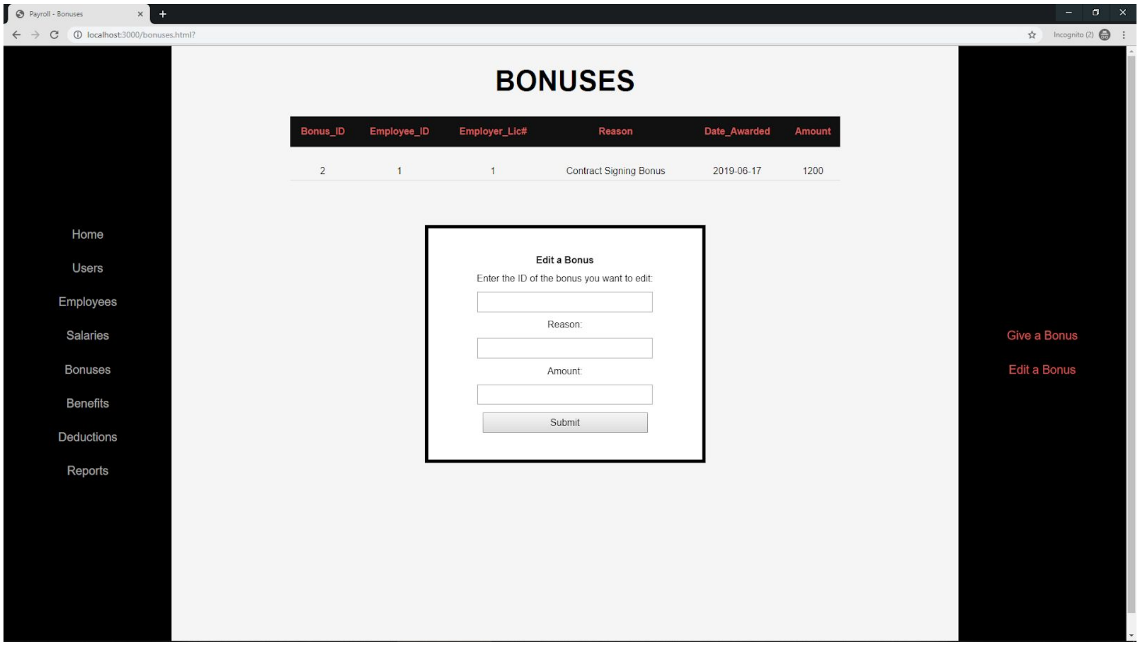
Recipient Employee ID:

Reason:

Amount:

Submit

Give a bonus



BONUSES

Bonus_ID	Employee_ID	Employer_Lic#	Reason	Date_Awarded	Amount
2	1	1	Contract Signing Bonus	2019-06-17	1200

Edit a Bonus

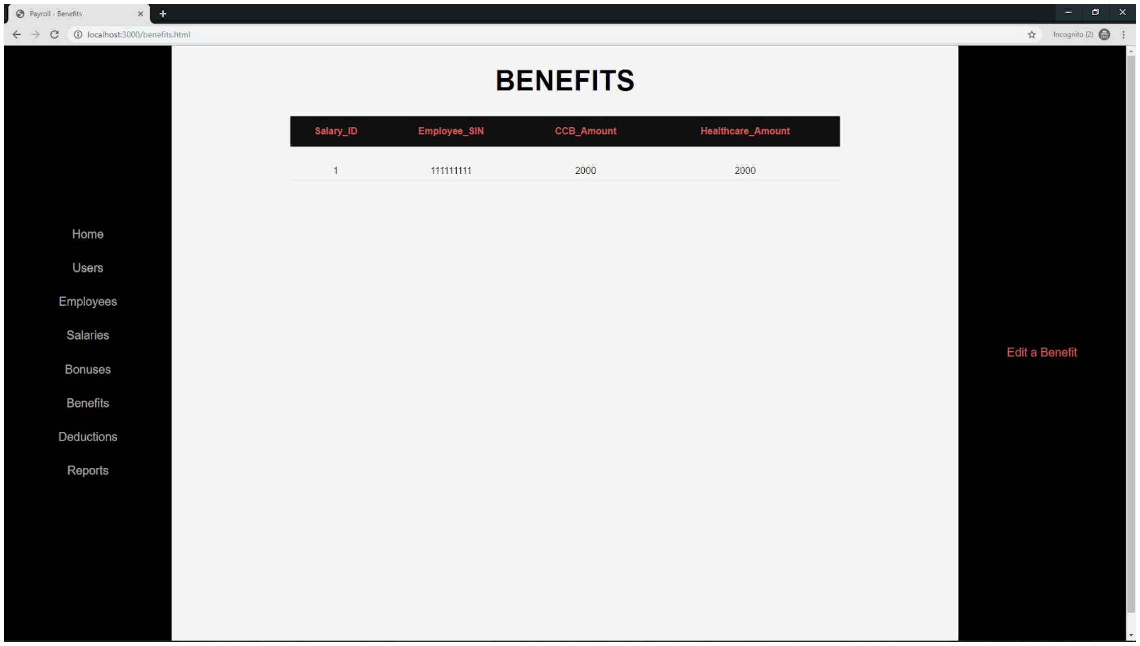
Enter the ID of the bonus you want to edit:

Reason:

Amount:

Edit a bonus

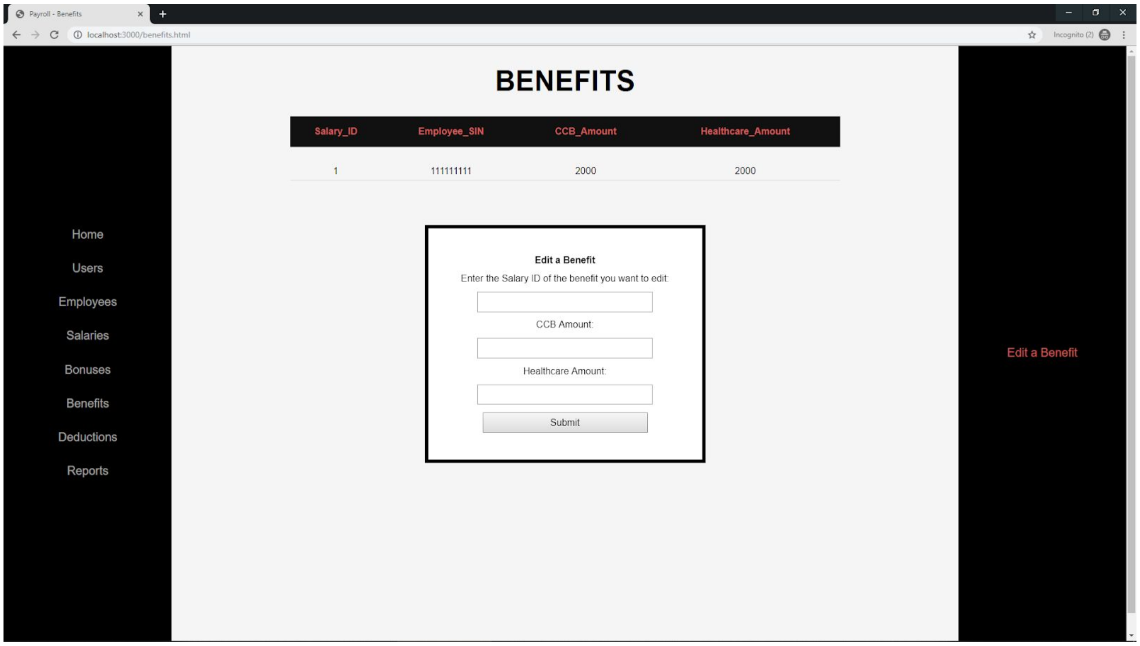
Benefits Page



BENEFITS

Salary_ID	Employee_SIN	CCB_Amount	Healthcare_Amount
1	111111111	2000	2000

Benefits are added automatically when salaries are assigned but you can use this page to edit them if you wish.



BENEFITS

Salary_ID	Employee_SIN	CCB_Amount	Healthcare_Amount
1	111111111	2000	2000

Edit a Benefit

Enter the Salary ID of the benefit you want to edit:

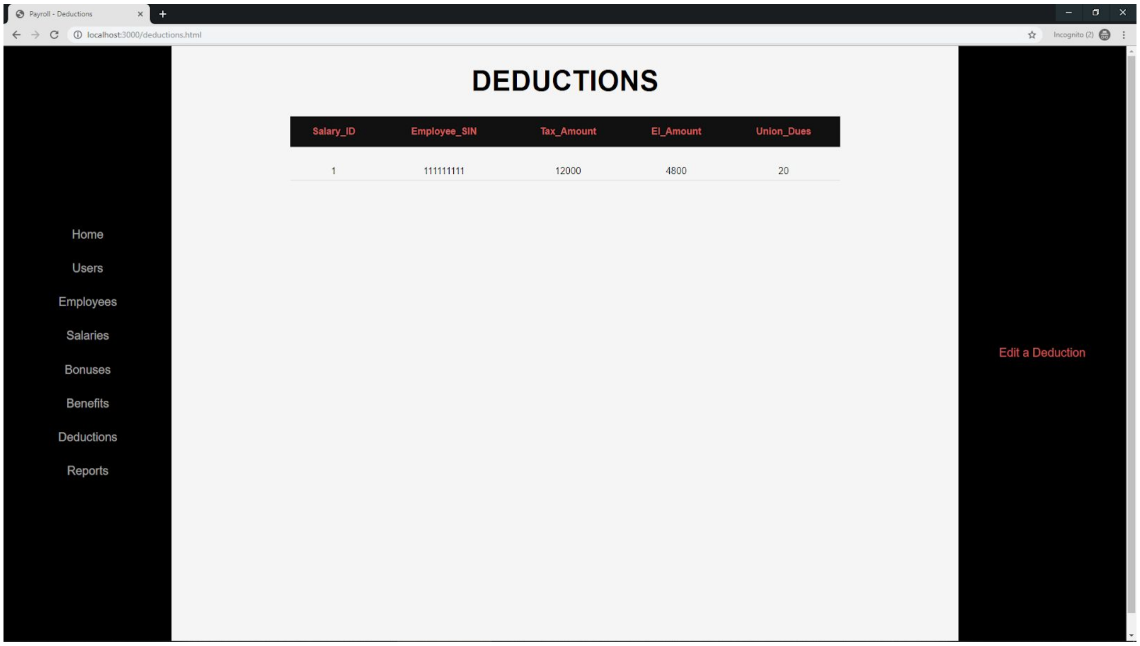
CCB Amount:

Healthcare Amount:

Submit

Edit a benefit

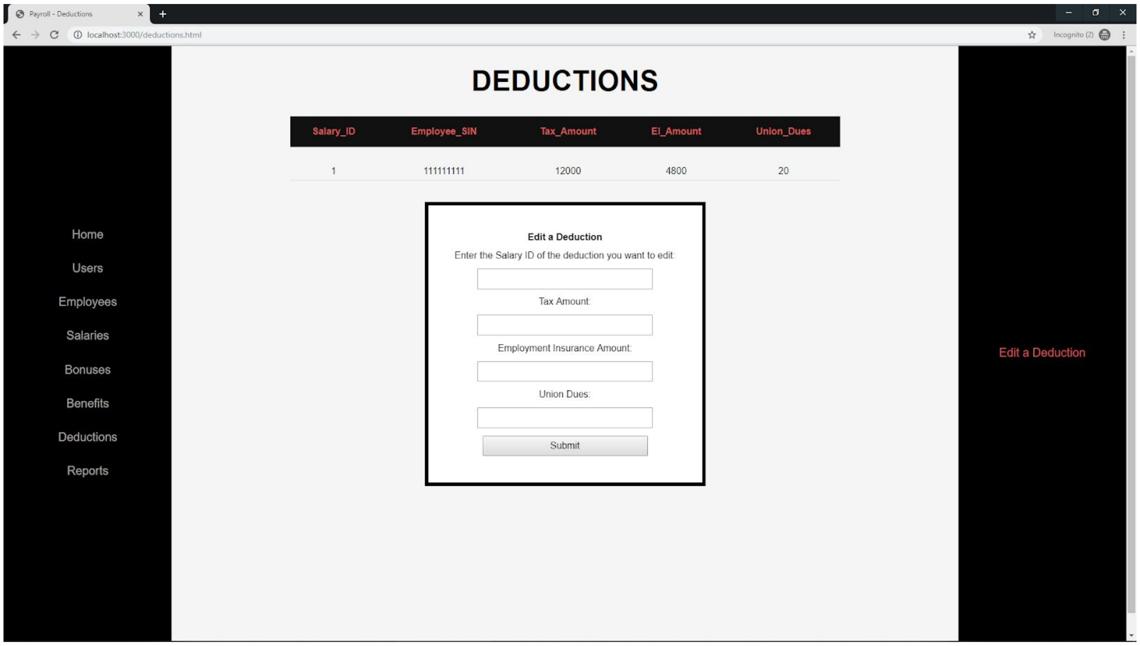
Deductions Page



DEDUCTIONS

Salary_ID	Employee_SIN	Tax_Amount	EI_Amount	Union_Dues
1	111111111	12000	4800	20

Deductions are added automatically when salaries are assigned but you can use this page to edit them if you wish.



DEDUCTIONS

Salary_ID	Employee_SIN	Tax_Amount	EI_Amount	Union_Dues
1	111111111	12000	4800	20

Edit a Deduction

Enter the Salary ID of the deduction you want to edit:

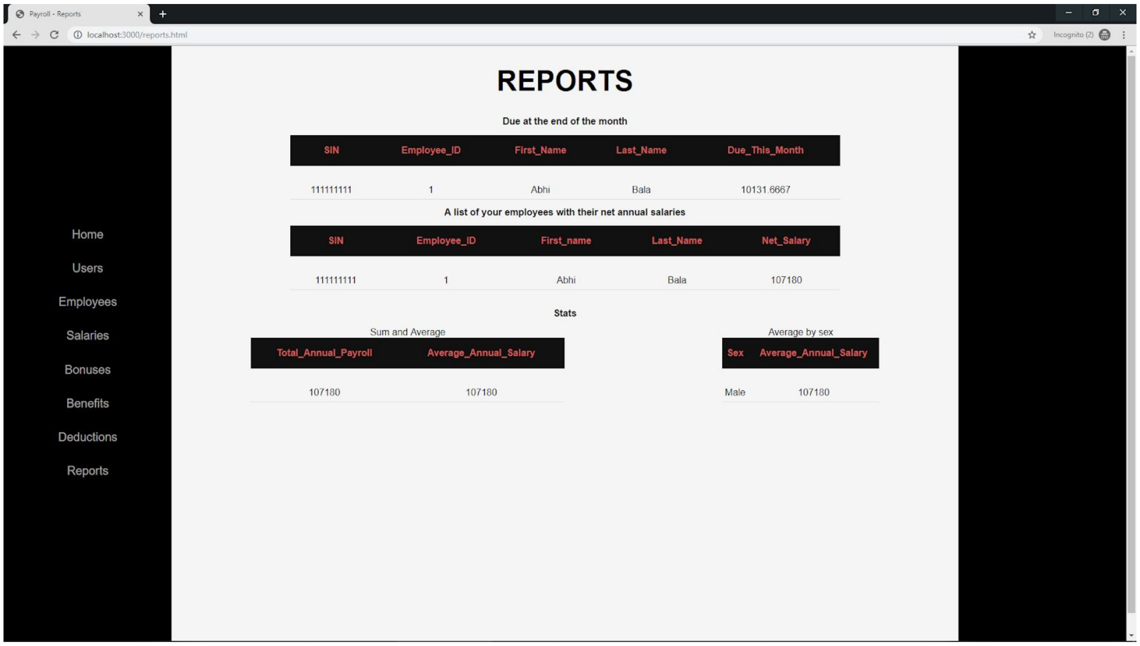
Tax Amount:

Employment Insurance Amount:

Union Dues:

Edit a deduction

Reports Page



REPORTS

Due at the end of the month

SIN	Employee_ID	First_Name	Last_Name	Due_This_Month
111111111	1	Abhi	Bala	10131.6667

A list of your employees with their net annual salaries

SIN	Employee_ID	First_name	Last_Name	Net_Salary
111111111	1	Abhi	Bala	107180

Stats

Sum and Average

Total_Annual_Payroll	Average_Annual_Salary
107180	107180

Average by sex

Sex	Average_Annual_Salary
Male	107180

Use this page to view reports about your company's payroll

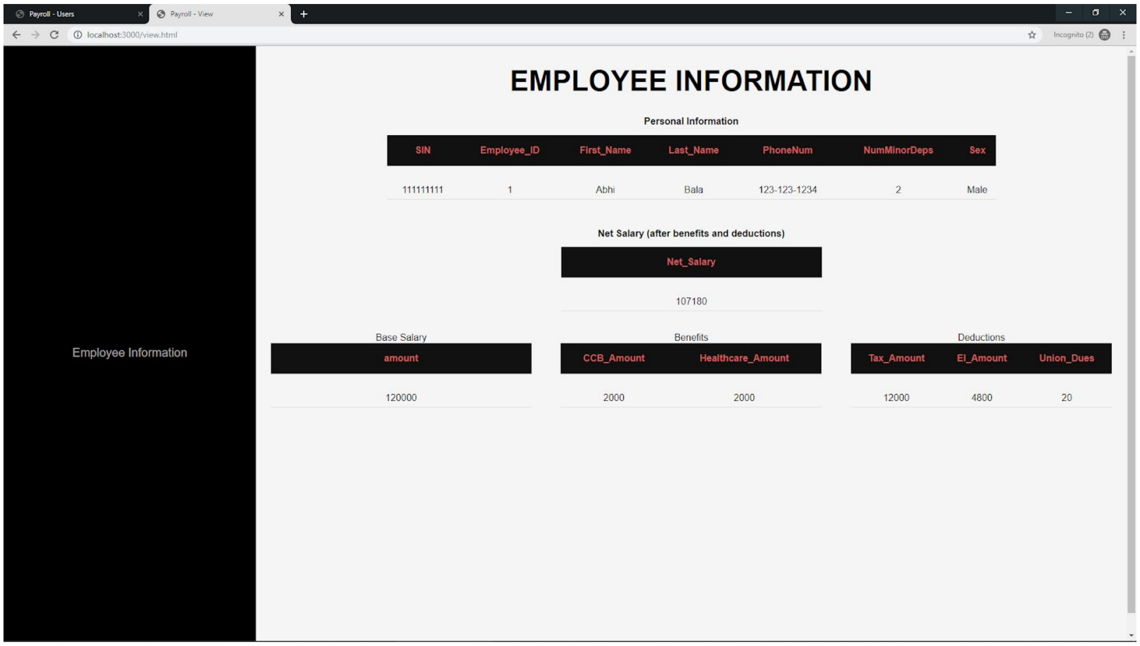
The first table indicates the monthly net salary of all employees in the system. This includes the base salary, all benefits, deductions, and bonuses awarded within the past month.

The second table lists the net annual salaries of all employees in the company.

The third table shows the total annual payroll of the company, as well as the average annual net salary of all the employees in the company.

The final table indicates the average annual net salary by sex.

Viewing the Site as an Employee



EMPLOYEE INFORMATION

Personal Information

SIN	Employee_ID	First_Name	Last_Name	PhoneNum	NumMinorDeps	Sex
111111111	1	Abhi	Bala	123-123-1234	2	Male

Net Salary (after benefits and deductions)

Net_Salary
107180

Base Salary

amount
120000

Benefits

CCB_Amount	Healthcare_Amount
2000	2000

Deductions

Tax_Amount	EI_Amount	Union_Dues
12000	4800	20

An employee only has access to the Employee Information page. Here, they can see:

- Their personal information
- Their net salary after benefits have been added and deductions have been taken away
- The constituent parts of their net salary
    - Their base salary
    - Their benefits
    - Their deductions

9. Conclusion

Overall, the project presented a unique learning opportunity to gain skills in much needed industry application of database design. From conceiving the initial entities and their attributes, to figuring out the relationships between them, then creating an entity relationship diagram and then producing a corresponding relational model created a good foundation to the project. With those steps completed, we looked to create queries and translate the work into actual implementation. Each step helped see the greater picture to the database clearly.

The course was brief but well designed, helping students to learn essential skills in database design. Thank you.

References

https://payroll.intuit.com/pricing/

https://clutch.co/press-releases/25-small-businesses-still-record-finances-paper-risking-errors-security-breaches

https://www.mysql.com/downloads/

https://dev.mysql.com/downloads/workbench/

Appendices

Appendix A: Company Table

CID	Name
97897	Group3 Inc.

Appendix B: User Table

SIN	First_Name	Last_Name	PhoneNum	CID	loginID	password
111111111	Abhi	Bala	123-123-1234	97897	3	728442
123456789	Hamzah	Umar	403-891-2571	97897	1	812557
222222222	Omar	Qureshi	123-123-0019	97897	5	211834
333333333	Kashfia	Sailunaz	123-019-3412	97897	6	172514
444444444	Jalal	Kawash	123-012-3412	97897	7	879057

Appendix C: Employer Table

LicenseNum	SIN
1	123456789

Appendix D: Employee Table

Employee_ID	SIN	NumMinorDeps	Sex
1	111111111	2	Male
3	222222222	4	Male
4	333333333	1	Female
5	444444444	3	Male

Appendix E: Salary Table

Salary_ID	Recipient_SIN	Employer_Lic#	Amount	Frequency
1	111111111	1	120000	Monthly
3	222222222	1	123000	Monthly
4	333333333	1	210000	Monthly
5	444444444	1	310000	Monthly

Appendix F: Bonus Table

Bonus_ID	Employee_ID	Employer_Lic#	Reason	Date_Awarded	Amount
2	1	1	Contract Signing Bonus	17-06-19	1200
3	3	1	Contract Signing Bonus	17-06-19	2000
4	4	1	Contract Signing Bonus	17-06-19	3000
5	5	1	Loyalty Bonus	17-06-19	4000

Appendix G: Benefit Table

Employee_SIN	Salary_ID	CCB_Amount	Healthcare_Amount
111111111	1	2000	2000
222222222	3	4000	2000
333333333	4	1000	2000
444444444	5	3000	2000

Appendix H: Deduction Table

Employee_SIN	Salary_ID	Tax_Amount	EI_Amount	Union_Dues
111111111	1	12000	4800	20
222222222	3	12300	4920	20
333333333	4	27300	10500	20
444444444	5	40300	15500	20