IT214 - DBMS

Case Study Database: Laborlist and Wages Database

Prepared by

Team No. 14 | Group 6 | Section 10

Group Members

Devdeep Shetranjiwala - 202001150

Devansh Patel - 202001262

Under the guidance of

Prof. Rachit Chhaya

TA - Vinay Maharaj

Index

Final Version of Software requirement specification

1. Purpose

The primary purpose of this document is to create a database for managing Data for LabourList and wages and build a system that will make the work of the Supervisors and Labors much easier. The purpose of this system will be to keep track of Laborers, their working hours and wages, type of work, minimum wage, rate per hour, payment status...etc.

The system also aims to save paper and cut down on time wastage. Along with these, users also value the system's ease of use, ability to create a paperless environment, simplicity of customization, and many other noteworthy features.

This database will provide fast and convenient access to the required data and allow Labors/Supervisors to view their profiles. It will allow Supervisors to keep track of the payment status and modify them as per their requirements. This document serves as a guide for developers of this management system.

2. Intended Audience and Reading Suggestions

This document is intended for database users like labor supervisors, laborers, and the city office of laborers.
Also, this database is helpful for various NGOs that are working for the betterment of Labor and are trying to increase their daily wages to help laborers. They can analyze this data and suggest some new changes/improvements.

3. Product Scope

The LabourList and wage management system will help laborers and their supervisors. It will help them to organize their data and work efficiently.

Also, the City's Municipal office will have an idea about the amount of work done in their respective city. It will help them to keep track of the progress of each work and Labour so that they can help them in every possible way.

Firms can also use this product, and with the help of this, they can track their progress on the work being done at their firm.

Also, Labor can track their wages and salaries weekly/hourly. They can track how much they have done in the last week. Also, they can add any suggestions/complaints they are facing during the work.

Banking details of each Labour will be in the database so that the Supervisor will pay directly to their respective bank account only, And it will create a transparent payment process.

Laborers can add some qualifications to their respective profiles, which will help them to get more work. Also, they can demand work in that specific domain itself.
Supervisors will give ratings to each of the laborers, and they will get bonuses accordingly.

4. Description

Product Perspective:

The labor list and wage database management system is a software application that will manage a large amount of data regarding the laborers and their work in a hierarchical way.

Product Functions:

The application will be updated daily to keep track of each laborer and their work. Also, we can put the entry and exit times of each labor. The payment can be made directly to the laborer via a digital medium to avoid any kind of corruption.

Government can keep track of the number of laborers in each city/district. This helps them to keep track of the current progress of the city.
Also, we can provide this to any individual firm for their use.

1. Add/Remove Laborer - New laborers can be added, and if someone leaves, we can remove that labor.
2. Add/Remove Supervisor - New Supervisor can be added, and if someone leaves, we can remove that labor.
3. Entry/Exit time - Supervisor will be able to update the entry and exit times of the laborers working under him.
4. Payment - The supervisor can pay the laborers using this functionality.
5. List of Laborers - The supervisor can access the details of all the laborers working under him.
6. List of Supervisors - Municipality department and the department of labor can access all the supervisors.
7. City Details - Municipality can track the current progress of the project with the help of work done.
8. Feedback - All the supervisors can view the feedback shared by laborers and work on that.

User Classes and Characteristics:

There will be four classes in which users can be classified.

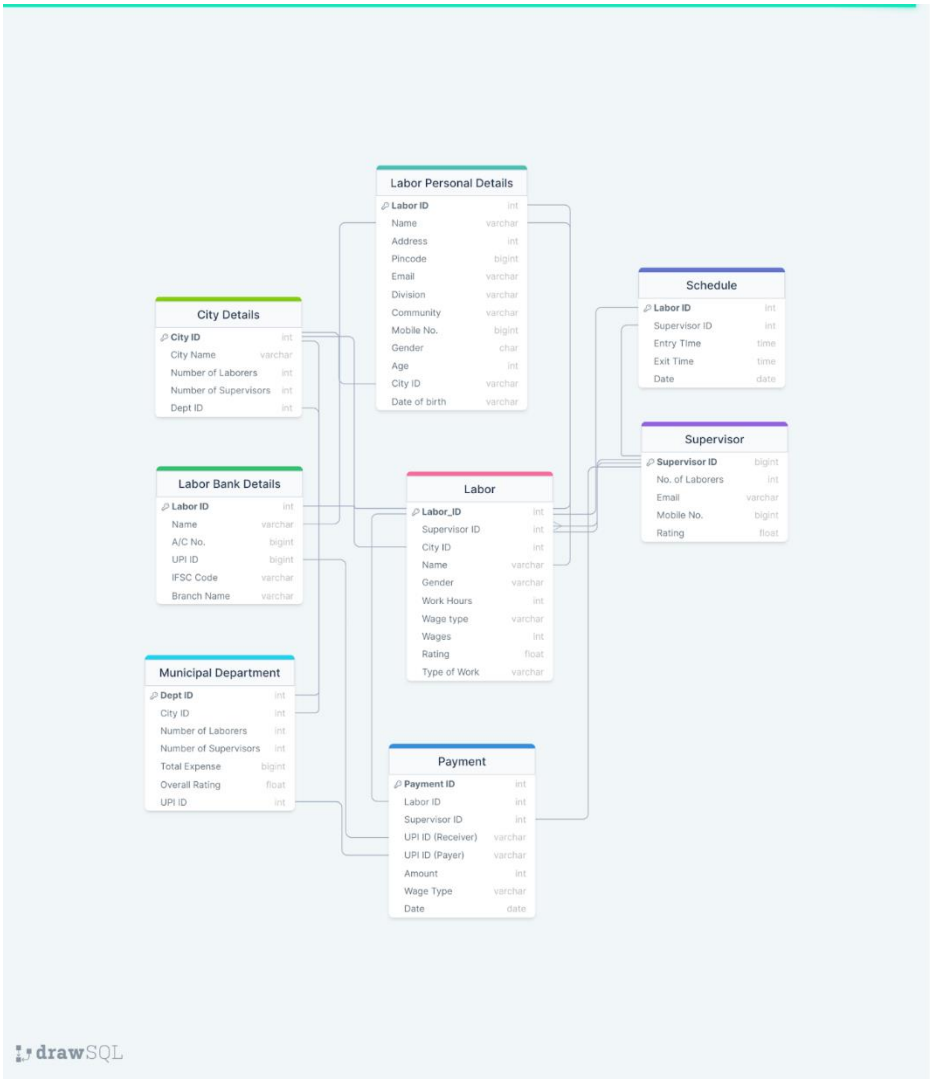
1. Labor
2. Labor Supervisor
3. Municipality office/Any other private firm
4. Department of Labor

Operating Environment:

As a web application, this system will be compatible with many other operating systems, including Windows, macOS, and Linux. A website will also be available for platforms like Android and iOS.

Relations to be defined in the Database:

Table Name	Table's Attributes
Labor	(**Labor ID**, Supervisor ID (F_Key), City ID (F_Key), Name, Gender, Work Hours, Wage type, Wages, Rating, Type of Work)
Payment	(**Payment ID,** Labor ID (F_Key), Supervisor ID (F_Key), UPI ID (receiver) (F_Key), UPI ID(payer)(F_Key), Amount, Wage Type (F_Key), Date)
Supervisor	(**Supervisor ID**, No. of Labors under Supervision, Email, Mobile No., Rating)
Schedule	(**Labor ID (F_key), Supervisor ID(F_key),** Entry Time, Exit Time, Date)
Labor Personal Details	(**Labor ID (F_key)**, Name, Address, Mobile Number, Gender, Date of birth, Age, Email, City ID, Pincode, Division, Community)
Municipal Department	(**Dept ID**, City ID, Number of Labors, Number of Supervisors, Total Expense, Overall Rating, UPI ID)
City details	(**City ID,** City Name, Number of Laborers, Number of Supervisors, Dept ID)
Labor Bank Details	(**Labor ID(F_Key),** Name (F_Key), A/C No., IFSC Code, Branch Name, UPI ID (receiver))



Real-World Work Flow of LabourList and wages:

- An existing laborer who has already registered will be given labor and wage according to his qualifications, experience, years, age, etc.
- There will be a separate view for laborers so that they can check out their details using the labor card and UUID given to them.
- If the laborer is new, they will be able to register and be assigned a UUID and a labor card by the labor supervisor.
- If work is done by labor, they will be paid according to workers' wages.
- These payments can be made online or offline depending on labor type and availability of online facilities.

- All the money transactions of a laborer are noted and stored and can be accessed by the labor, city office of labor, and labor supervisor.

2. Document the Requirements Collection/ Fact-Finding Phase

1. Background Reading/s

There are many Labor Management Systems present in the market. It consists of a good number of features for the employees. Personal information, attendance, performance history, expenses, timesheets, relieving letters, holidays, ID proofs, and other data could be stored in database management systems.

1. Zimyo.com

   This database has some very important and helpful features like storage of HR data, Employee Tracking, Custom Fields, Integration with HRMS Tools, etc.

   But, this database management system mainly focuses on the employees of large Multinational companies. In this new employee, the integration goes smoothly.

2. Serpent.com

   provides functionalities like geo-tracking. Attendance tracking, Attendance regularization, shift scheduling, bulk attendance, import attendance, biometric integration, and so on.

   These databases do not focus on the laborers like plumbers, electricians, or transport. workers, or house cleaning staff.

   In addition to that, there is no central Authority to manage this data.

Individual firms handle their data. And it is very tedious for the government to track each firm's data. Also, firms can easily manipulate the data.

3. ilo.org

India Wage Report Wage policies for decent work and inclusive growth provide functionalities like daily wages of laborers, avg wages in India, the gender gap wages, wages by sector, wages by occupation, wages by the level of education, wages by social background, wages by state, wages by inequality.

Different sector-wise details like agriculture, construction, economic sector primary, secondary, tertiary and class regular workers, casual workers, urban, rural and, male, female gender wages ratio.

These types of databases do focus on the laborers like Legislators, senior officials and managers, Professionals, Technicians, and associate professionals, Clerks, Service workers and shop and market sales workers, Skilled agricultural and fishery workers, Craft and related trades workers, Plant and machine operators and assemblers, Elementary occupations by dividing this occupation in diff 9 division.

In addition, there is a central Authority to manage this data.

Individual firms handle their data, but firms cannot easily manipulate it, which gives a sense of reliability.

ii. References

Zimyo.com

https://www.zimyo.com/insights/employee-database-management-system/

Serpentcs.in

https://www.serpentcs.in/product/labour-outsourcing-management-system

www.ilo.org

https://www.ilo.org/wcmsp5/groups/public/---asia/---ro-bangkok/---sro-new_delhi/documents/publication/wcms_638305.pdf

iii. List the combined Requirements gathered from Background Reading/s.

We require a database that can help the laborers, not just the employees of the firms. The integrated payment system will help them get their wages directly into their bank account after completing their work. And also it reduces the chance of corruption in the system. This system will help supervisors with the entry and exit time of each labor and helps them to calculate their wages automatically.

2. Interviews

1.

Interview Plan:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/09

Interviewee: 1) Santosh Pandor Designation: Supervisor

Interviewer: 1) Devansh Patel Designation: Developer

Date: 28/09/2022

Duration: 30 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the supervisors in their workplace and some suggestions from their side.

Agenda:

Problem with the current payroll system
Difficulties in managing data for each employee

Documents to be brought to the interview:

Rough plan of building
Any documents relating to current wage management procedures

Interview Summary:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/09
Interviewee: 1) Santosh Pandor Designation: Supervisor
Interviewer: 1) Devansh Patel Designation: Developer
Date: 28/09/2022
Duration: 30 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the laborers in their workplace and some suggestions from their side.

Result of the interview:

After the interview, we got to know that the payment to each employee daily is a very time-consuming process and very difficult to handle when the number of laborers is higher.

It is challenging to note the entry and exit time of each of the laborers, and there is a chance that the pen-paper thing goes missing.

2.

Interview Plan:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/10
Interviewee: 1) Ustadji Designation: Labor

Interviewer: 1) Devdeep Shetranjiwala Designation: Developer
Date: 28/09/2022
Duration: 30 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the laborers in their workplace and some suggestions from their side.

Agenda:

Problem with the current payroll system
Difficulties in managing data for each employee

Documents to be brought to the interview:

Rough plan of building
Any documents relating to current wage management procedures
Pen-paper and a rough estimate for a laborer.

Interview Summary:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/10
Interviewee: 1) Ustadji Designation: Labor
Interviewer: 1) Devdeep Shetranjiwala Designation: Developer
Date: 28/09/2022
Duration: 30 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the laborers in their workplace and some suggestions from their side.

Result of the interview:

Sometimes, there is a delay in the payment from the supervisor's side. In the modern era, cash handling is complex, and everyone prefers online payment.

3.

Interview Plan:

Municipality Department: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/11
Interviewee: 1) Mr. Kumar Designation: Labor Officer
Interviewer: 1) Devansh Patel Designation: Developer
Date: 30/09/2022
Duration: 45 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the city municipality office in their workplace and some suggestions from their side.

Agenda:

Problem with the current Laborlist/payroll system
Difficulties in managing data for each employee

Documents to be brought to the interview:

Rough plan of building
Any documents relating to current wage management procedures
Current labor management chart

Interview Summary:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/11
Interviewee: 1) Mr. Kumar Designation: Labor officer
Interviewer: 1) Devansh Patel Designation: Developer
Date: 30/09/2022
Duration: 45 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the city municipality department in their workplace and some suggestions from their side.

Result of the interview:

There are lot of paperwork going on for the laborers' appointments and track for their work, and it is challenging to manage all the laborers appointed by the city municipality department. Also, some supervisors do not give proper wages to their laborers and put that extra money in their pockets.

In addition to that, it is tough and challenging for them to keep track of the progress of the work assigned to laborers because the municipality department has to follow some challenging deadlines.

4.

Interview Plan:

Municipality Department: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/12
Interviewee: 1) Mr. Jay Shah Designation: Minister, Department of Labor
Interviewer: 1) Devdeep Shetranjiwala Designation: Developer
Date: 30/09/2022
Duration: 60 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the Department of labor in their workplace and some suggestions from their side.

Agenda:

Problem with the current Laborlist/payroll system
Difficulties in managing data for each employee

Documents to be brought to the interview:

Rough plan of building
Any documents relating to current wage management procedures
Appointment letter

Interview Summary:

Shinestar Enterprise: (Role Play) Interview Plan

Project Reference: SF/SJ/2022/12

Interviewee: 1) Mr. Jay Shah Designation: Minister, Department of Labor

Interviewer: 1) Devdeep Shetranjiwala Designation: Developer

Date: 30/09/2022

Duration: 60 Min Location: DA-IICT.

Purpose of Interview:

To identify the issues faced by the Department of labor in their workplace and some suggestions from their side.

Result of the interview:

The Department of labor manages many things regarding the laborers and their wages. They calculate the minimum wage for the laborers of various departments according to the hard work and the difficulty of that particular job.

They do not have proper data regarding each job to calculate the minimum wages and improve the condition of laborers. Also, the ministry wants to reduce corruption in the current wages management system. And they want to digitize each and everything for transparency purposes.

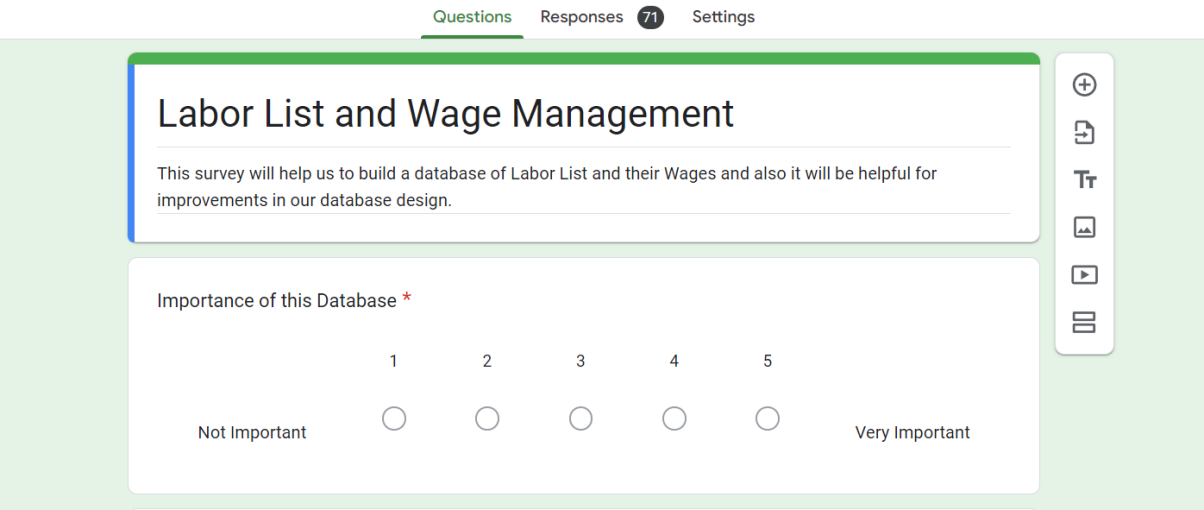
Combined Requirements gathered from the Interviews

- The payment to each employee daily is a very time-consuming process and difficult to handle when the number of laborers is higher.
- It is challenging to note the entry and exit time of each of the laborers.

- Sometimes, there is a delay in the payment from the supervisor's side.
- In the modern era, cash handling is complex, and everyone prefers online payment.
- Also, corruption handling is a challenge when payments are done offline
- and are not under the direct supervision of the city office of laborers.

3. Questionnaires

We have created a google form and circulated it to students, laborers, and supervisors, mainly from DA-IICT and food stalls and shops near DA-IICT.



Importance of this Database *

	1	2	3	4	5	
Not Important	○	○	○	○	○	Very Important

Do you believe that there is a chance of corruption in our current Wage System? *

- ○ Yes
- ○ No

Which type of wage you prefer for laborers? *

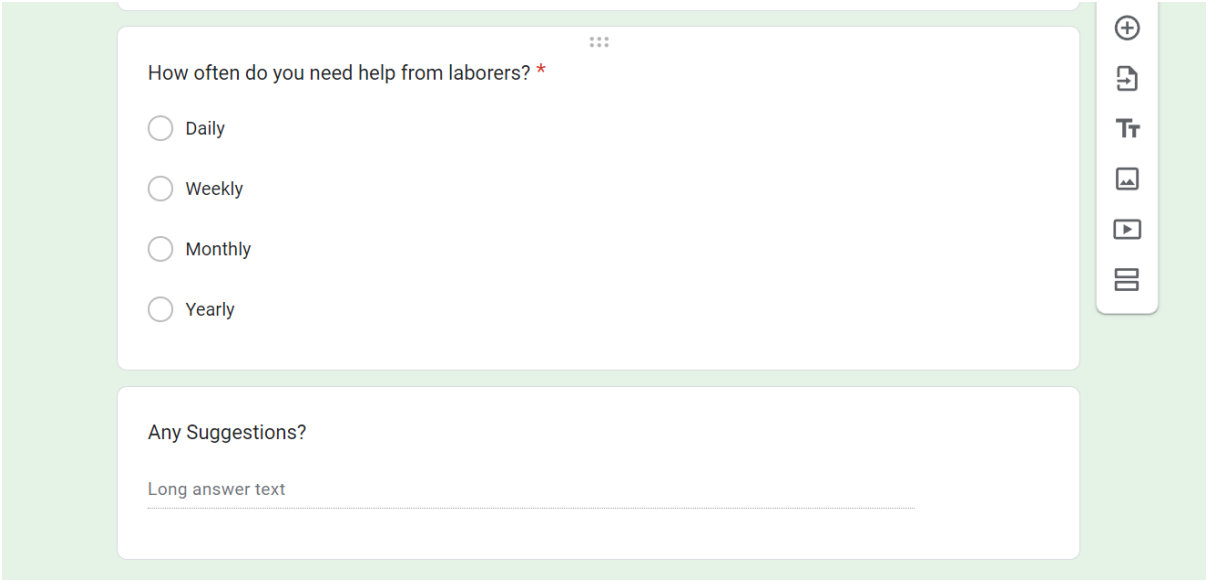
- ○ Weekly Wage
- ○ Hourly Wage

Which type of payment method would be preferred by laborers? *

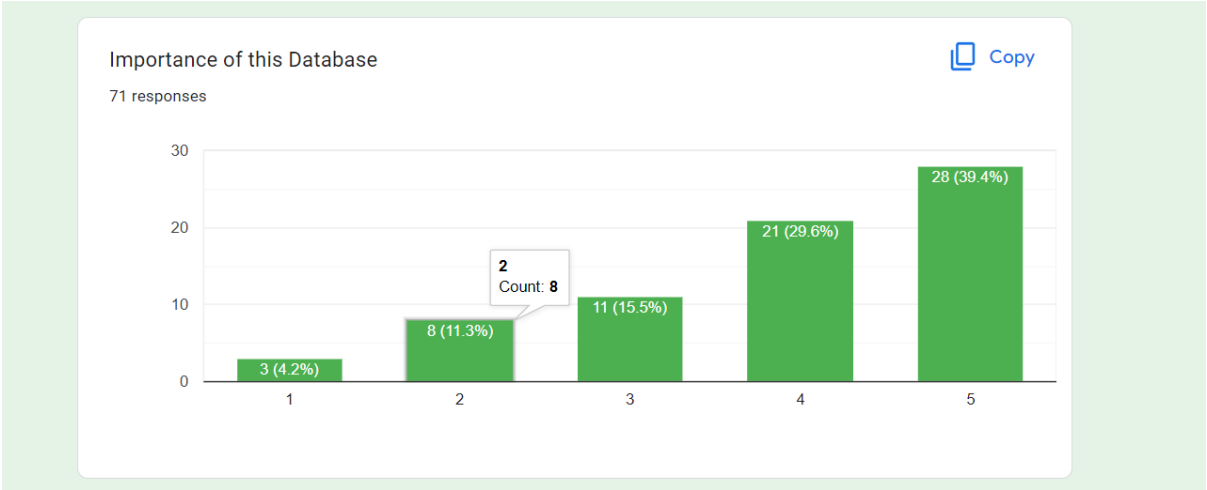
- ○ Cash
- ○ Online

What are categories of labor work you deal with on the regular basis? *

- ☐ Plumber
- ☐ Electrician
- ☐ Cleaning staff
- ☐ Transport
- ☐ Construction



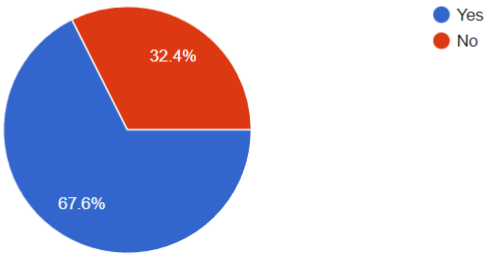
The responses are given below:



Do you believe that there is a chance of corruption in our current Wage System?

Copy

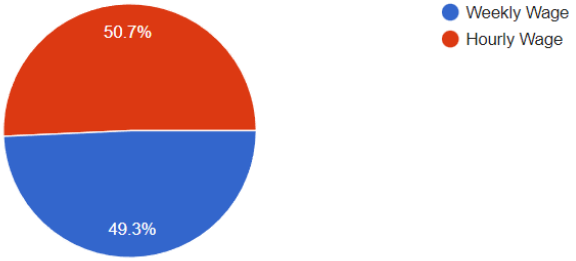
71 responses



Which type of wage you prefer for laborers?

Copy

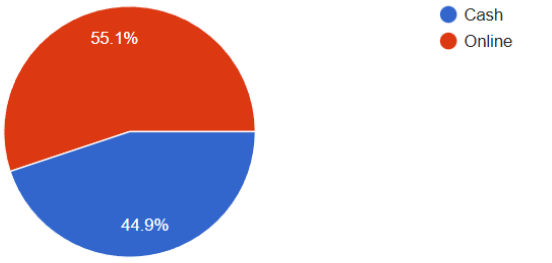
71 responses



Which type of payment method would be preferred by laborers?

Copy

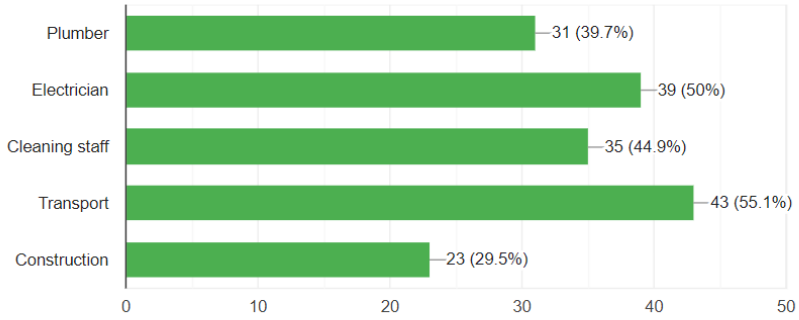
78 responses



What are categories of labor work you deal with on the regular basis?

Copy

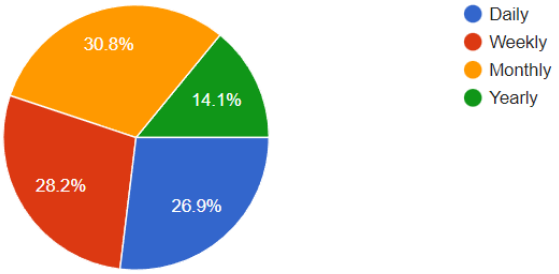
78 responses



How often do you need help from laborers?

Copy

78 responses



Any Suggestions?

4 responses

This thing can help a lot.

Too much corruption now days.

Try to increase daily wages to help laborers.

Try to digitalize whole system.

Observations based on the answers to the questionnaire.

- As we can see from the responses, people take help from different types of laborers.

- There is an opinion difference in the payment system, whether to keep it online or cash.
- People pay differently to the laborers, like weekly/hourly.
- The responses show high demand for this database management system.

4. Observation

Shinestar Enterprise

Project Reference: SF/SJ/2022/13
Observations by: Devansh Patel
Date: 30/09/2022 **Time:** 14:30
Duration: 45 minutes **Place:** Shinestar Enterprise

Observations:

- Porper system for entry and exit is required.
- It is difficult for supervisors to calculate daily wage at a particular instance and give the laborers on the spot.
- When the number of laborers increases, it is difficult for the supervisors and the owner to manage.
- Cash dealing is complicated at the time of exit.
- Government and the municipality do not have any proper way to count the number of laborers currently working.
- Paper wastage is too much in the current process.

List of combined requirements from Observations:

- The database should have a proper system to encounter the entry and exit times of each laborer.
- An online payment system is needed for the laborers who work hourly.

- The Department of labor must have accurate data of every employer to provide them benefits.
- An online LabourList and wage management system will reduce many paper and human hours.
- Database should be able to handle a large number of laborers and supervisors.
- Databases must have some kind of recovery system; if a crash occurs, we should be able to recover the previous data without any hassle.

3. Fact-Finding Chart

Objective	Technique	Subject(s)	Time commitment
To get a background on the current LaborList and Wage Management System	Background Reading (Study current systems on the Web)	Company Reports, Blogs, Websites	one day
To understand the current working of the labor management system and how the payments are made	Interview, Observation, Questionnaires	Labor Supervisor	45 Minutes
To gain an understanding of the current system from the labor perspective	Interview/ Observations	Labor	30 Minutes
To find out the requirements from the municipality department	Interview	Labor Supervisor, 2 District labor officers	One day

To find out new features to be added in the system	Interviews/ Reading	1 Labor, 1 Supervisor, 1 District labor officer	One day

4. List Requirements

- Automatic payment system for the laborers.
- Port interface for supervisors to track the entry and exit of each of the laborers.
- Rating system for both laborers and supervisors.
- Data should be organized to help the municipality department to track the progress of the work assigned.
- Crash recovery system in the application to prevent data loss in case of a crash.
- Payment should be made automatically by calculating the hours worked multiplied by the hourly wage if the laborer is working hourly.
- Comment/Suggestion window must be provided to the laborers if required.
- Separate page for government and various NGOs for the betterment of laborers and supervise each laborer and the supervisor.
- Updation of the database must be regular to keep track of the real-world scenario.

5. User Categories and Privileges

I.Laborers

Access Privileges: Relations: Labor, Schedule, Labor Personal Details, Labor Bank Details

Labor can have all the information provided by him as well as the payment status, work done by his/her, entry and exit time, the rating provided by the supervisor, etc.

II. Supervisors

Access Privileges: Relations: Labor, Schedule, Payment, Supervisor

Supervisor can access all the related information about the laborers assigned to him. The supervisor will make payments to all the related laborers, so he/she will have access to the payment table as well. Supervisors will access the rating system, which will provide ratings to their respective laborers.

III. Municipality office/Any other private firm

Access Privileges: Relation: Labor, Supervisor, Payment, Labor Personal Details, City Details

Municipality department will have access to all the laborers and supervisors. They can access all the payments made by the supervisors to their laborers. They can track the overall work done in the city. They will check the ratings of the supervisors and laborers. And assign them to work respectively, and change their wages also.

IV. Department of Labor

Access Privileges: Relation: Labor, Supervisor, Labor Personal Details, City Details, Municipality Department, Labor Bank Details

Department of labor will maintain all the records related to laborers and their supervisors. They will monitor every transaction done to the laborers to keep the system transparent and efficient. They will check individual cities' ratings in terms of development

and guide their respective municipality departments for better development. They will also monitor the daily wages given to the laborers, and they will increase the daily wages of laborers if required after discussing with the NGOs and by viewing their ratings and personal details.

6. Assumptions

For the system to work efficiently, we must assume certain things.

- Supervisors and laborers will update their entry and exit timing regularly.
- The Enduser must fulfill some hardware requirements to run this application smoothly.
- All the ratings provided to the supervisors and laborers should be genuine.
- There should not be any false entries on laborers' part to gain more wages.

7. Business Constraints

- All the labor assignments and their payment are real-time. The labor allocation and wage payment process should occur instantly and be updated in real-time to ensure the smooth running of the LaborList and Wage Management.
- The available labor, laborer, hardware, and computing power are limited.
- The capability of every laborer to accept online payment is limited means some laborers are not aware or not capable of accepting online payment.

References:

Zimyo.com
https://www.zimyo.com/insights/employee-database-management-system/

Serpentcs.in
https://www.serpentcs.in/product/labour-outsourcing-management-system
www.ilo.org
https://www.ilo.org/wcmsp5/groups/public/---asia/---ro-bangkok/---sro-new_delhi/documents/publication/wcms_638305.pdf
IEEE Template
https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

Noun Analysis

Noun and Verb Analysis

Serial no.	Nouns	Verbs
1	labor	Pay
2	Age	Make
3	Supervisor	Work
4	Department	Keep
5	Gender	See
6	Wage	Check
7	Wage Type	Manipulate
8	Labor class	Assign
10	Schedule	Consisting
11	Municipal Department	Remove
12	Labor type	Update
13	Web Application	Sort
14	Division	Scope
15	Developer	Manage
16	City	Implement
17	Education	Feedback
18	Cast	Register
19	City ID	Completed
20	Payment	Be
21	Rating	List

22	Email	Have
23	Mobile	Can
24	Address	Supervising
25	Amount	Modify
26	Community	Demanding
27	UPI ID	Agree
28	Bank Details	Admit
29	Pincode	Attend
30	IFSC code	Ask
31	A/C No.	Compare
32	Entry Time	Examine
33	Exit Time	Develop
34	Total Time	Fetch
35	Payer	Ignore
36	Receiver	Include
37	Date of birth	Sign
38	Hours	Will
39	Daily	Records
40	Weekly	Store
41	Urban	Geo-Tracking
42	Rural	Alert
43	Admin	Communicate
44	Database	Delete
45	Management	Add

46	Payment history	Trace
47	Labor Name	Hire
48	UUID	Note
49	Front End	Reimburse
50	Back End	Refurbish
51	Postgresql	
52	Developer ID	
53	Payment ID	
54	Supervisor ID	
55	Developer Name	
56	VS Code	
57	E-R Diagram	
58	Economics	
59	Experience	
60	History	
61	ID Proofs	
62	Holidays	
63	HR Data	
64	NGOs	
65	Ministers	
66	Casual Workers	
67	Clarks	
68	Technicians	
69	Interview	

70	No of laborers	
71	No of supervisors	
72	Payroll system	
73	B-Tree	
74	Social Background	
75	Gender Wage Ratio	
76	Class Regular Workers	
77	Economic Sector	
78	Agriculture	
79	APIs	
80	Authority	
81	References	
82	Product Scope	
83	Business Model	
84	Transparent System	
85	Primary Key	
86	Unique Key	
87	Forign Key	
88	Attendance	
89	Biometric	
90	Location	
91	Live	
92	Maps	
93	Labor Activity	

94	Real world	
95	Status	
96	Cash	
97	Online	
98	Payment Status	
99	Labor card	
100	Service	

Noun and Verb Analysis

Accepted Noun and verbs List

Candidate Entity set	Candidate entry set	Candidate relationship set
Labor	Labor ID, Supervisor ID, City ID, Name, Gender, Work Hours, Wage type, Wages, Rating, Type of Work	Work, accept, register, feedback, give, complaint, fill
Payment	Payment ID, Labor ID , Supervisor ID, UPI ID (receiver) , UPI ID(payer), Amount, Wage Type, Date	Transfer, verify, add, remove.
Supervisor	Supervisor ID, No. of Labors under Supervision, Email, Mobile No., Rating	Assign, pay, management, give rating, Delete, add, create
Schedule	Labor ID, Supervisor ID, Entry Time, Exit Time, Date	Change, make, edit

Labor Personal Details	Labor ID, Name, Address, Mobile Number, Gender, Age, Email, City ID, Pincode, Division, Community	Update, add, delete
Municipal Department	Dept ID, City ID, Number of Labors, Number of Supervisors, Total Expense, Overall Rating, UPI ID	Observe, records, alert
City details	City ID, City Name, Number of Laborers, Number of Supervisors, Dept ID	Manage, observe
Labor Bank Details	Labor ID, Name, A/C No., IFSC Code, Branch Name, UPI ID (receiver)	Pay, update, add, remove

Rejected Noun & Verb List

Nouns	Reject Reason
Web Application	General
Developer	General
Total Time	Attribute
Date of birth	Vague
Hours	General
Daily	Associations
Weekly	Associations
Urban	Irrelevant
Rural	Irrelevant

Admin	General
Database	Vague
Management	Vague
Payment history	Duplicate
Labor Name	Attribute
UUID	Attribute
Front End	Vague
Back End	Vague
Postgresql	Duplicate
Developer Name	Vague
VS Code	Vague
E-R Diagram	General
Economics	Vague
Experience	General
Holidays	Vague
HR Data	General
NGOs	Vague
Ministers	Vague
Casual Workers	General
Clarks	General
Technicials	General
Interview	Vague
No of laborers	Associations
No of supervisors	Associations

Payroll system	Duplicate
B-Tree	Vague
Social Background	General
Gender Wage Ratio	Irrelevant
Class Regular Workers	Irrelevant
Economic Sector	General
Agriculture	General
APIs	Vague
Authority	Duplicate
References	Vague
Product Scope	Irrelevant
Business Model	General
Transparent System	General
Primary Key	Irrelevant
Unique Key	Irrelevant
Forign Key	Irrelevant
Attendance	General
Biometric	Duplicate
Location	General
Live	General
Maps	Irrelevant
Labor Activity	Vague
Real world	Irrelevant
Status	Duplicate

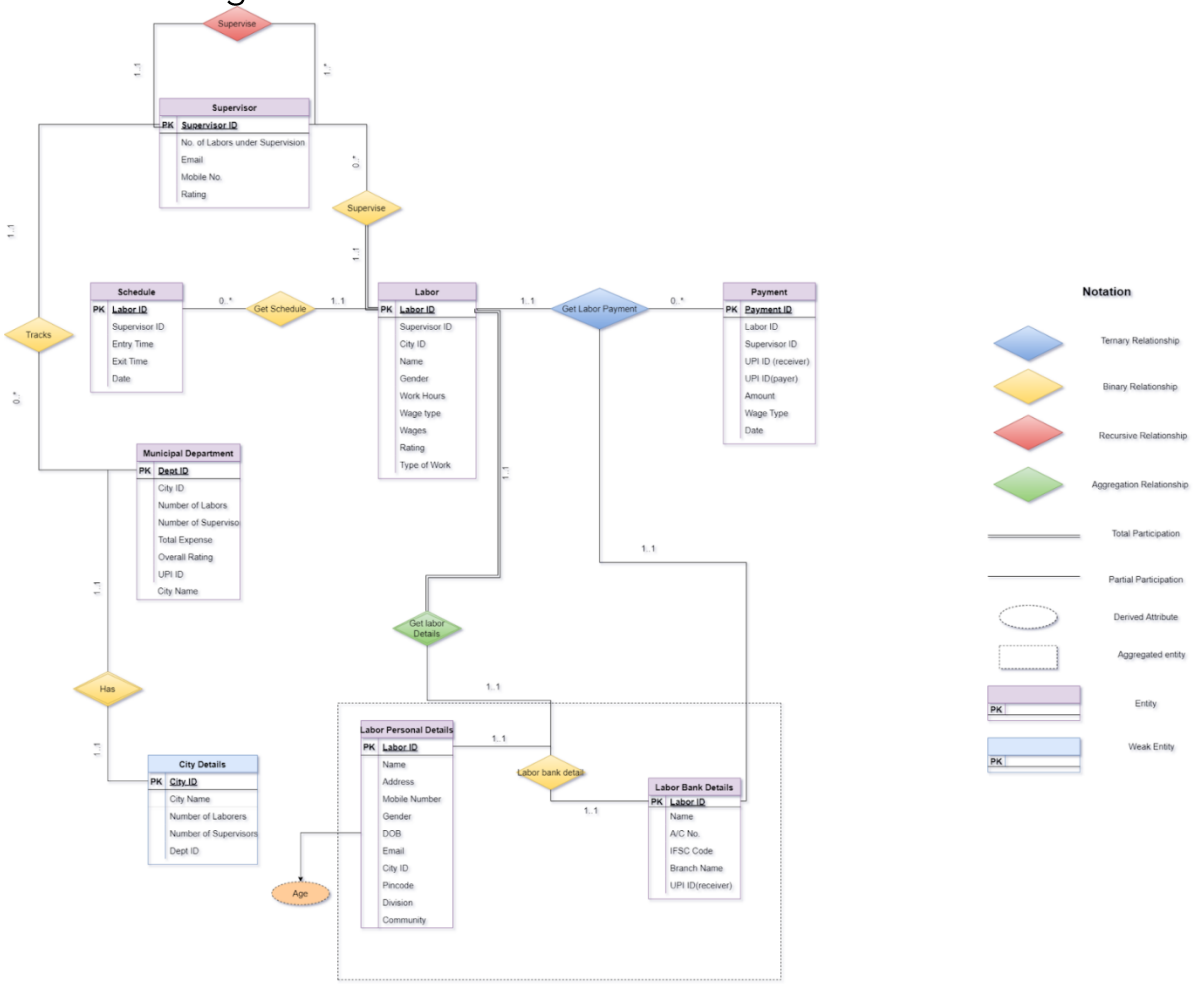
Cash	Vague
Online	Vague
Payment Status	Duplicates
Labor card	Irrelevant
Service	Vague

Verbs	Reject Reason
Keep	Vague
See	Vague
Check	General
Manipulate	Duplicate
Consisting	General
Update	Duplicate
Sort	General
Scope	Vague
Implement	General
Register	General
Completed	Vague
Be	Vague
List	General
Have	Vague
Can	Vague
Modify	General
Demanding	Vague

Agree	General
Admit	General
Attend	General
Ask	General
Compare	Vague
Examine	Vague
Develop	General
Fetch	Vague
Ignore	Vague
Include	Duplicate
Sign	General
Will	General
Records	Vague
Store	Vague
Geo-Tracking	General
Alert	Vague
Communicate	Vague
Delete	Duplicate
Trace	Duplicate
Hire	General
Note	Vague
Reimburse	Duplicate
Refurbish	Duplicate

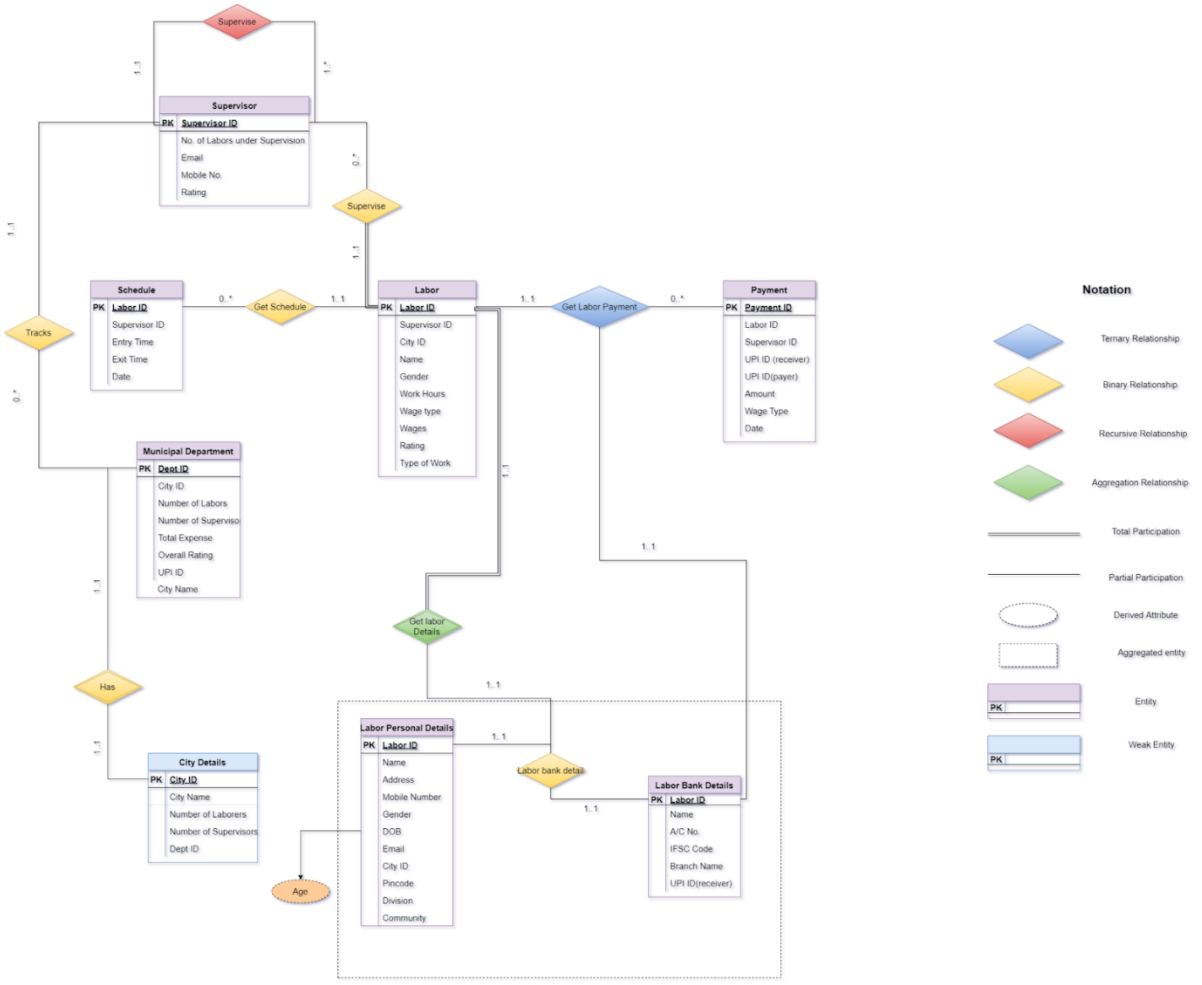
E-R Diagram:

V1 of E-R Diagram

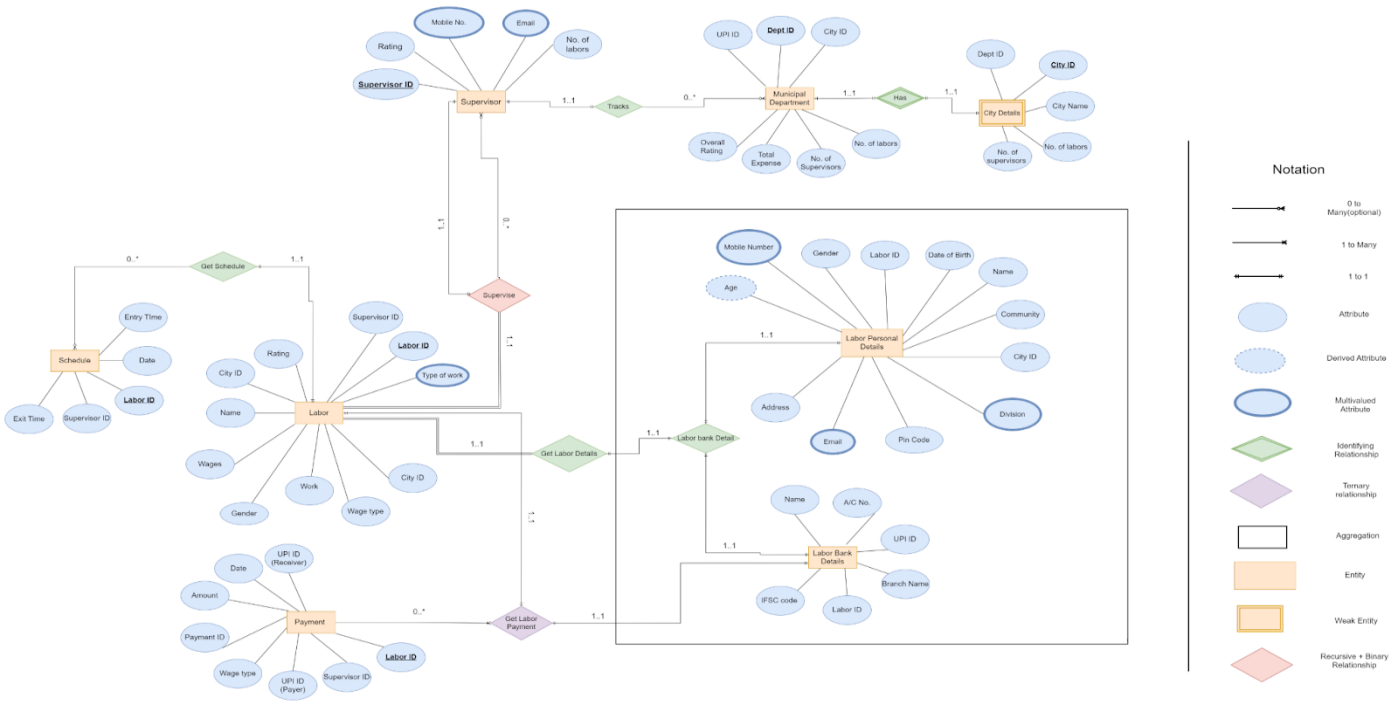


Final Version of E-R diagram

Class diagram



Conceptual/Physical E-R Diagram



City Details is a weak entity.

Type of relations shown in the final version of E-R diagram

Recursive Relationship - Supervisor can supervise other supervisors.
Aggregation - Aggregation in get labor details and get labor bank details.
Ternary Relationship - get labor is a ternary relationship.

Total Participation is between Entity Labor and Relationship Get labor details, Entity labor and relationship supervise.

Multivalued Attribute - Email ID, mobile number.
Derived Attribute - Age from Date of Birth

Relationship with Schema

1. Labor (**Labor ID**, Supervisor ID, Rating, City ID, Name, Wages, Gender, Work, Wage Type)
2. Payment (**Labor ID**, **Date,**UPI ID(Receiver), Amount, Payment ID, Wage Type, UPI ID(Payer), Supervisor ID)
3. Schedule (**Labor ID**, Date, Entry Time, Exit Time, Supervisor ID)
4. Supervisor (**Supervisor ID**, Rating, No. of Laborers)
5. Supervisor_mobile (**Supervisor ID**, Mobile No.)
6. Supervisor_email (**Supervisor ID**, email)
7. Municipal Department (**Dept ID**, City ID, No. of Laborers, No. of Supervisors, Overall Rating, Total Expense, UPI ID)
8. City Details (**City ID**, Dept ID, City Name, No. of Laborers, no. of Supervisors)
9. Labor Bank Details (**UPI ID**, Name, Branch Name, Labor ID, IFSC Code, A/C No.)
10. Labor Type of Work (**Labor ID, Type of Work**)
11. Labor Personal Details (**Labor ID**, Age, Pin Code, Gender, Date of Birth, Name, Community, City ID)
12. Labor Mobile no. (**Labor ID, Mobile No.**)
13. Labor Division (**Labor ID, Division**)
14. Labor Email (**Labor ID, Email**)
15. Labor Details (**Labor ID**, UPI ID)

Functional Dependencies

1. Labor (Labor ID → Name, Gender, Rating, Work, Wage ; Wage type → Wage)

2. Payment(Labor ID → UPI ID(Receiver), Date, Amount, Payment ID, Wage Type, UPI ID(Payer), Supervisor ID; Supervisor ID→ UPI ID(Payer))

3. Schedule (Labor ID → Date, Entry time, Exit time, Supervisor ID)

4. Supervisor (Supervisor ID→ Rating, No. of Labors)

5. Supervisor_mobile (Supervisor ID→ Mobile No.)

6. Supervisor_email (Supervisor ID → email)

7. Municipal Department (Dept ID → City ID, No. of Laborers, No. of Supervisors, Overall Rating, Total Expense, UPI ID; City ID → No. of Laborers, No. of Supervisors; UPI ID→ Total Expense)

8. City Details (City ID →Dept ID, City Name, No. of Laborers, no. of Supervisors)

9. Labor Bank Details (UPI ID → Name, Branch Name, Labor ID, IFSC Code, A/C No. ; A/C No→ Branch Name, IFSC Code)

10. Labor Type of Work: no dependencies

11. Labor Personal Details (Labor ID → Age, Pin Code, Gender, Date of Birth, Name, Community, City ID; Date of Birth →Age)

12. Labor Mobile no.: No Dependencies

13. Labor Division: No Dependencies

14. Labor Email: no dependencies

15. Labor Details: (Labor ID → UPI ID)

Redundancy and Analysis:

1. Labor (**Labor ID**, Supervisor ID, Rating, City ID, Name, Wages, Gender, Work, Wage Type)
    - Here Table is already in 2NF form.
    - There is a transitive relationship between Labor ID and Wages.

2. Payment (**Labor ID**, **Date,**UPI ID(Receiver), Amount, Payment ID, Wage Type, UPI ID(Payer), Supervisor ID)
    - Here Table is already in 2NF form.
    - There is a transitive relationship between Supervisor ID and UPI ID(Payer).

3. Schedule (**Labor ID**, Date, Entry Time, Exit Time, Supervisor ID)
    - There are no dependencies.
    - Table is already in 3NF.

4. Supervisor (**Supervisor ID**, Rating, No. of Laborers)
    - Here Table is already in 3NF form.
    - There are no dependencies.

5. Supervisor_mobile (**Supervisor ID**, Mobile No.)
    - Here Table is already in 3NF form.
    - There are no dependencies.

6. Supervisor_email (**Supervisor ID**, email)
    - Here Table is already in 3NF form.
    - There are no dependencies.

7. Municipal Department (**Dept ID**, City ID, No. of Laborers, No. of Supervisors, Overall Rating, Total Expense, UPI ID)
    - The table is already in 2NF.

- There is a transitive relationship between Dept ID and Total Expenses.
- There are transitive relationships between City ID and No. of laborers, City ID, and No. of supervisors.

8. City Details (**City ID**, Dept ID, City Name, No. of Laborers, no. of Supervisors)
    - Here Table is already in 3NF form.
    - There are no dependencies.

9. Labor Bank Details (**UPI ID**, Name, Branch Name, Labor ID, IFSC Code, A/C No.)
    - The table is already in 2NF.
    - There is a transitive relation between UPI ID & Branch Name and UPI ID & IFSC Code.

10. Labor Type of Work (**Labor ID, Type of Work**)
    - Here Table is already in 3NF form.
    - There are no dependencies.

11. Labor Personal Details (**Labor ID**, Age, Pin Code, Gender, Date of Birth, Name, Community, City ID)
    - The table is already in 2NF.
    - There are no redundancies.

12. Labor Mobile no. (**Labor ID, Mobile No.**)
    - Here Table is already in 3NF form.
    - There are no dependencies.

13. Labor Division (**Labor ID, Division**)
    - Here Table is already in 3NF form.
    - There are no dependencies.

14. Labor Email (**Labor ID, Email**)
    - Here Table is already in 3NF form.
    - There are no dependencies.

15. Labor Details (**Labor ID**, UPI ID)
    - Here Table is already in 3NF form.
    - There are no dependencies.

3NF/BCNF Normalization:

1. Labor (**Labor ID**, Supervisor ID, Rating, City ID, Name, Gender, Work, Wage Type)
   LaborWage (**Wage type, Wage**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only one attribute (Labor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- We have formed a separate table for wage type and wage to remove the transitive dependency. So now it's in 3NF form.

2. Payment (**Labor ID**, UPI ID(Receiver), Date, Amount, Payment ID, Wage Type, UPI ID(Payer))
   Payer_detail(**UPI ID(Payer), Supervisor ID**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only one attribute (Labor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- We have formed a table for wage type and wage to remove the transitive depen
- dency. So now it's in 3NF form.

3. Schedule (**Labor ID**, **Date**, Entry Time, Exit Time, Supervisor ID)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only one attribute (Labor ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form, which implies it is in 3NF form.

4. Supervisor (**Supervisor ID**, Rating, No. of Laborers)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Supervisor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

5. Supervisor_mobile (**Supervisor ID**, **Mobile No.**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Supervisor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

6. Supervisor_email (**Supervisor ID, email**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Supervisor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.

- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

7. Municipal Department (**Dept ID**, City ID, Overall Rating, Total Expense, UPI ID)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Dept_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- To remove transitive dependency, we have removed No. of supervisors and No. of laborers attribute from tables, So now it's in 3NF form.

8. City Details (**City ID**, Dept ID, City Name, No. of Laborers, no. of Supervisors)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (City_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

9. Labor Bank Details (**UPI ID**, Name, Labor ID, A/C No.)
Labor Account Details(**A/C No.**, IFSC code, Branch Name)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (UPI_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.

- To remove transitive dependency we have formed a separate table for A/C No, IFSC Code, Branch Name. So now it's in 3NF form.

10. Labor Type of Work (**Labor ID, Type of Work**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There are only 2 attributes (Labor ID, Type of Work) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

11. Labor Personal Details (**Labor ID**, Age, Pin Code, Gender, Date of Birth, Name, Community, City ID)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Labor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form. .

12. Labor Mobile no. (**Labor ID, Mobile No.**)

- This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
- There are only 2 attributes(Labor ID, Mobile No.) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.

- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

13. Labor Division (**Labor ID, Division**)

    - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
    - There are only 2 attributes (Labor_ID,Division) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
    - All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

14. Labor Email (**Labor ID, Email**)

    - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
    - There are no partial dependencies. Hence, it is in 2NF form.
    - All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

15. Labor Details (**Labor ID**, UPI ID)

    - This schema does not have any composite or multivalued attributes. It satisfies atomicity. Hence, it is already in 1NF form.
    - There is only 1 attribute (Labor_ID) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
    - All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

Final Relationship with schema:

1. Labor (**Labor ID**, Supervisor ID, Rating, City ID, Name, Gender, Work, Wage Type)
    - Foreign Key Supervisor ID refers to Supervisors.

2. LaborWage (**Wage type, Wage**)

3. Payment (**Labor ID, Date**, UPI ID(Receiver), Amount, Payment ID, Wage Type, UPI ID(Payer))

4. Payer_detail(**UPI ID(Payer), Supervisor ID**)

5. Schedule (**Labor ID**, Date, Entry Time, Exit Time, Supervisor ID)
    - Foreign Key Supervisor ID refers to Supervisors.
    - Foreign Key Labor ID refers to Labor.

6. Supervisor (**Supervisor ID**, Rating, No. of Laborers)

7. Supervisor Mobile (**Supervisor ID, Mobile No.**)
    - Foreign Key Supervisor ID refers to Supervisors.

8. Supervisor Email (**Supervisor ID, Email**)
    - Foreign Key Supervisor ID refers to Supervisors.

9. Municipal Department (**Dept ID**, City ID, Overall Rating, Total Expense, UPI ID)
    - Foreign Key City ID refers to City details.

10. City Details (**City ID**, Dept ID, City Name, No. of Laborers, No. of Supervisors)
    - Foreign Key Dept ID refers to the Municipal Department.

11. Labor Bank Details (**UPI ID**, Name, Labor ID, A/C No.)
    - Foreign Key Labor ID refers to Labor.

12. Labor Account Details(A/C No., IFSC code, Branch Name)
    - Foreign Key A/C No. refers to Labor Bank Details.

13. Labor Type of Work (Labor ID, Type of Work)
    - Foreign Key Labor ID refers to Labor.

14. Labor Personal Details (Labor ID, Pin Code, Gender, Date of Birth, Name, Community, City ID)

15. Labor Mobile no. (Labor ID, Mobile No.)
    - Foreign Key Labor ID refers to Labor.

16. Labor Division (Labor ID, Division)
    - Foreign Key Labor ID refers to Labor.

17. Labor Email (Labor ID, Email)
    - Foreign Key Labor ID refers to Labor.

18. Labor Details (Labor ID, UPI ID)
    - Foreign Key Labor ID refers to Labor.

DDL Scripts:

```
-- Table: public.City Details

-- DROP TABLE IF EXISTS public."City Details";

CREATE TABLE IF NOT EXISTS public."City Details"
(
    "City ID" bigint NOT NULL,
    "Dept ID" bigint NOT NULL,
    "City Name" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Number of Laborers" bigint,
    "Number of Supervisors" bigint,
    CONSTRAINT "City Details_pkey" PRIMARY KEY ("City
ID"),
    CONSTRAINT "City Details_Dept ID_key" UNIQUE
("Dept ID"),
    CONSTRAINT "City Details_Dept ID_fkey" FOREIGN
KEY ("Dept ID")
        REFERENCES public."Municipal Department"
("Dept ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."City Details"
    OWNER to postgres;


-- Table: public.Labor

-- DROP TABLE IF EXISTS public."Labor";

CREATE TABLE IF NOT EXISTS public."Labor"
(
    "Labor ID" bigint NOT NULL,
    "Supervisor ID" bigint NOT NULL,
    "Rating" bigint NOT NULL,
    "City ID" bigint,
    "Name" character varying COLLATE
pg_catalog."default" NOT NULL,
```

```
    "Gender" character varying COLLATE
pg_catalog."default",
    "Work" character varying COLLATE
pg_catalog."default",
    "Wage Type" character varying COLLATE
pg_catalog."default",
    CONSTRAINT "Labor_pkey" PRIMARY KEY ("Labor ID"),
    CONSTRAINT "Labor_City ID_fkey" FOREIGN KEY
("City ID")
        REFERENCES public."City Details" ("City ID")
MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Supervisor ID" FOREIGN KEY
("Supervisor ID")
        REFERENCES public."Supervisor" ("Supervisor
ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor"
    OWNER to postgres;
```

-- Table: public.Labor Account Details

```
-- DROP TABLE IF EXISTS public."Labor Account Details
";

CREATE TABLE IF NOT EXISTS public."Labor Account
Details "
(
    "A/C No." character varying COLLATE
pg_catalog."default" NOT NULL,
    "IFSC code" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Branch Name" character varying COLLATE
pg_catalog."default",
```

```
    CONSTRAINT "Labor Account Details _pkey" PRIMARY
KEY ("A/C No.")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Account Details "
    OWNER to postgres;
```

-- Table: public.Labor Bank Details

```
-- DROP TABLE IF EXISTS public."Labor Bank Details";

CREATE TABLE IF NOT EXISTS public."Labor Bank
Details"
(
    "Name" character varying COLLATE
pg_catalog."default" NOT NULL,
    "A/C No." character varying COLLATE
pg_catalog."default",
    "UPI ID" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Labor ID" bigint NOT NULL,
    CONSTRAINT "Labor Bank Details_pkey" PRIMARY KEY
("UPI ID"),
    CONSTRAINT "Labor Bank Details_Labor ID_fkey"
FOREIGN KEY ("Labor ID")
        REFERENCES public."Labor" ("Labor ID") MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Bank Details"
    OWNER to postgres;
```

-- Table: public.Labor Details

```
-- DROP TABLE IF EXISTS public."Labor Details";

CREATE TABLE IF NOT EXISTS public."Labor Details"
(
    "Labor ID" bigint NOT NULL,
    "UPI ID" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Details_pkey" PRIMARY KEY
("Labor ID"),
    CONSTRAINT "Labor Details_Labor ID_fkey" FOREIGN
KEY ("Labor ID")
        REFERENCES public."Labor" ("Labor ID") MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Labor Details_UPI ID_fkey" FOREIGN
KEY ("UPI ID")
        REFERENCES public."Labor Bank Details" ("UPI
ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Details"
    OWNER to postgres;
```

-- Table: public.Labor Divison

```
-- DROP TABLE IF EXISTS public."Labor Divison ";

CREATE TABLE IF NOT EXISTS public."Labor Divison "
(
    "Labor ID" bigint NOT NULL,
    "Division" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Divison _pkey" PRIMARY KEY
("Labor ID", "Division")
        INCLUDE("Division"),
```

```
    CONSTRAINT "Labor Divison _Labor ID_fkey" FOREIGN
KEY ("Labor ID")
        REFERENCES public."Labor Personal Details"
("Labor ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Divison "
    OWNER to postgres;
```

-- Table: public.Labor Email

```
-- DROP TABLE IF EXISTS public."Labor Email";

CREATE TABLE IF NOT EXISTS public."Labor Email"
(
    "Labor ID" bigint NOT NULL,
    "Email" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Email_pkey" PRIMARY KEY ("Labor
ID", "Email")
        INCLUDE("Email"),
    CONSTRAINT "Labor Email_Labor ID_fkey" FOREIGN
KEY ("Labor ID")
        REFERENCES public."Labor Personal Details"
("Labor ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Email"
    OWNER to postgres;
```

-- Table: public.Labor Mobile no.

```
-- DROP TABLE IF EXISTS public."Labor Mobile no.";

CREATE TABLE IF NOT EXISTS public."Labor Mobile no."
```

```
(
    "Labor ID" bigint NOT NULL,
    "Mobile no." bigint NOT NULL,
    CONSTRAINT "Labor Mobile no._pkey" PRIMARY KEY
("Labor ID", "Mobile no.")
        INCLUDE("Mobile no."),
    CONSTRAINT "Labor Mobile no._Labor ID_fkey"
FOREIGN KEY ("Labor ID")
        REFERENCES public."Labor Personal Details"
("Labor ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Mobile no."
    OWNER to postgres;
```

-- Table: public.Labor Personal Details

```
-- DROP TABLE IF EXISTS public."Labor Personal
Details";

CREATE TABLE IF NOT EXISTS public."Labor Personal
Details"
(
    "Gender" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Labor ID" bigint NOT NULL,
    "Date of Birth" date,
    "Name" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Age" bigint,
    "Community" character varying COLLATE
pg_catalog."default",
    "City ID" bigint NOT NULL,
    "Pin code" bigint,
    "Address" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Personal Details_pkey" PRIMARY
KEY ("Labor ID")
```

```
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor Personal Details"
    OWNER to postgres;
```

-- Table: public.Labor type of work

```
-- DROP TABLE IF EXISTS public."Labor type of work";

CREATE TABLE IF NOT EXISTS public."Labor type of
work"
(
    "Labor ID" bigint NOT NULL,
    "Type of Work" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor type of work_pkey" PRIMARY KEY
("Labor ID", "Type of Work")
        INCLUDE("Type of Work"),
    CONSTRAINT "Labor type of work_Labor ID_fkey"
FOREIGN KEY ("Labor ID")
        REFERENCES public."Labor" ("Labor ID") MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Labor type of work"
    OWNER to postgres;
```

-- Table: public.LaborWage

```
-- DROP TABLE IF EXISTS public."LaborWage";

CREATE TABLE IF NOT EXISTS public."LaborWage"
(
    "Wage type" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Wage" character varying COLLATE
pg_catalog."default" NOT NULL,
```

```
    CONSTRAINT "LaborWage_pkey" PRIMARY KEY ("Wage
type", "Wage")
        INCLUDE("Wage")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."LaborWage"
    OWNER to postgres;
```

-- Table: public.Municipal Department

```
-- DROP TABLE IF EXISTS public."Municipal
Department";

CREATE TABLE IF NOT EXISTS public."Municipal
Department"
(
    "UPI ID" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Dept ID" bigint NOT NULL,
    "City ID" bigint NOT NULL,
    "Overall Rating" double precision NOT NULL,
    "Total Expense" bigint NOT NULL,
    CONSTRAINT "Municipal Department_pkey" PRIMARY
KEY ("Dept ID"),
    CONSTRAINT "Municipal Department_City ID_fkey"
FOREIGN KEY ("City ID")
        REFERENCES public."City Details" ("City ID")
MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Municipal Department"
    OWNER to postgres;
```

-- Table: public.Payer_detail

```
-- DROP TABLE IF EXISTS public."Payer_detail";
```

```
CREATE TABLE IF NOT EXISTS public."Payer_detail"
(
    "UPI ID (Payer) " character varying COLLATE
pg_catalog."default" NOT NULL,
    "Supervisor ID" bigint NOT NULL,
    CONSTRAINT "Payer_detail_pkey" PRIMARY KEY ("UPI
ID (Payer) ", "Supervisor ID")
        INCLUDE("Supervisor ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Payer_detail"
    OWNER to postgres;
-- Table: public.Payment

-- DROP TABLE IF EXISTS public."Payment";

CREATE TABLE IF NOT EXISTS public."Payment"
(
    "UPI ID (Receiver)" character varying COLLATE
pg_catalog."default" NOT NULL,
    "UPI ID (Payer)" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Date" date NOT NULL,
    "Amount" bigint,
    "Payment ID" bigint,
    "Wage Type" character varying COLLATE
pg_catalog."default" NOT NULL,
    "Labor ID" bigint NOT NULL,
    CONSTRAINT "Payment_pkey" PRIMARY KEY ("Date",
"Labor ID")
        INCLUDE("Date")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Payment"
    OWNER to postgres;


-- Table: public.Schedule

-- DROP TABLE IF EXISTS public."Schedule";
```

```
CREATE TABLE IF NOT EXISTS public."Schedule"
(
    "Entry Time" time with time zone NOT NULL,
    "Exit Time" time with time zone NOT NULL,
    "Supervisor ID" bigint NOT NULL,
    "Labor ID" bigint NOT NULL,
    "Date" date NOT NULL,
    CONSTRAINT "Schedule_pkey" PRIMARY KEY ("Labor
ID", "Date")
        INCLUDE("Date"),
    CONSTRAINT "Schedule_Entry Time_key" UNIQUE
("Entry Time"),
    CONSTRAINT "Schedule_Exit Time_key" UNIQUE ("Exit
Time"),
    CONSTRAINT "Schedule_Labor ID_fkey" FOREIGN KEY
("Labor ID")
        REFERENCES public."Labor" ("Labor ID") MATCH
SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "Supervisor ID" FOREIGN KEY
("Supervisor ID")
        REFERENCES public."Supervisor" ("Supervisor
ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Schedule"
    OWNER to postgres;


-- Table: public.Supervisior_mobile

-- DROP TABLE IF EXISTS public."Supervisior_mobile";

CREATE TABLE IF NOT EXISTS
public."Supervisior_mobile"
(
    "Supervisior ID" bigint NOT NULL,
```

```
    "Mobile No." bigint NOT NULL,
    CONSTRAINT "Supervisior Mobile no._pkey" PRIMARY
KEY ("Supervisior ID", "Mobile No.")
        INCLUDE("Mobile No."),
    CONSTRAINT "Supervisior Mobile no._Supervisior
ID_fkey" FOREIGN KEY ("Supervisior ID")
        REFERENCES public."Supervisor" ("Supervisor
ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Supervisior_mobile"
    OWNER to postgres;
```

-- Table: public.Supervisor

```
-- DROP TABLE IF EXISTS public."Supervisor";

CREATE TABLE IF NOT EXISTS public."Supervisor"
(
    "Supervisor ID" bigint NOT NULL,
    "Rating" double precision NOT NULL,
    "Number of Laborers" bigint,
    "Mobile Number" bigint,
    "Email" character varying COLLATE
pg_catalog."default",
    CONSTRAINT "Supervisor_pkey" PRIMARY KEY
("Supervisor ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Supervisor"
    OWNER to postgres;
```

-- Table: public.Supervisor_email

```
-- DROP TABLE IF EXISTS public."Supervisor_email";
```

```
CREATE TABLE IF NOT EXISTS public."Supervisor_email"
(
    "Supervisior ID" bigint NOT NULL,
    "Email" character varying COLLATE
pg_catalog."default" NOT NULL,
    CONSTRAINT "Supervisor Email_pkey" PRIMARY KEY
("Supervisior ID", "Email")
        INCLUDE("Email"),
    CONSTRAINT "Supervisor Email_Supervisior ID_fkey"
FOREIGN KEY ("Supervisior ID")
        REFERENCES public."Supervisor" ("Supervisor
ID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public."Supervisor_email"
    OWNER to postgres;
```

DDL Snapshot:

Table: Supervisor

Query | Query History

```
-- Table: LaborList and Wages.Supervisor

-- DROP TABLE IF EXISTS "LaborList and Wages"."Supervisor";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Supervisor"
(
    "Supervisor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Rating " double precision NOT NULL,
    "No. of Laborers " bigint,
    CONSTRAINT "Supervisor_pkey" PRIMARY KEY ("Supervisor ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Supervisor"
    OWNER to postgres;



SELECT * FROM "LaborList and Wages"."Supervisor"
```

Table: City Details

```
-- Table: LaborList and Wages.City Details

-- DROP TABLE IF EXISTS "LaborList and Wages"."City Details";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."City Details"
(
    "City ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Dept ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "City Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "No. of Laborers " bigint,
    "No. of Supervisors" bigint,
    CONSTRAINT "City Details_pkey" PRIMARY KEY ("City ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."City Details"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."City Details"
```

Table: Municipal Department

Query | Query History

```sql
-- Table: LaborList and Wages.Municipal Department

-- DROP TABLE IF EXISTS "LaborList and Wages"."Municipal Department";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Municipal Department"
(
    "Dept ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "City ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Overall Rating" double precision,
    "Total Expense" bigint,
    "UPI ID(Payer)" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Municipal Department_pkey" PRIMARY KEY ("Dept ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Municipal Department"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Municipal Department"
```

Data output | Messages | Notifications

Dept ID [PK] character varying	City ID character varying	Overall Rating double precision	Total Expense bigint	UPI ID(Payer) character varying

Table: Labor

Query | Query History

```sql
-- Table: LaborList and Wages.Labor

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Supervisor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Rating" double precision,
    "City ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Gender" character varying COLLATE pg_catalog."default" NOT NULL,
    "Work" character varying COLLATE pg_catalog."default",
    "Wage Type" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor_pkey" PRIMARY KEY ("Labor ID")
)
Loading...
TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Labor"
```

Data output | Messages | Notifications

Labor ID [PK] character varying	Supervisor ID character varying	Rating double precision	City ID character varying	Name character varying	Gender character varying	Work character varying	Wage Type character varying

Table: Labor Account Details

Query | Query History

```
1  -- Table: LaborList and Wages.Labor Account Details
2
3  -- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Account Details";
4
5  CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Account Details"
6  (
7      "A/C No." character varying COLLATE pg_catalog."default" NOT NULL,
8      "IFSC Code" character varying COLLATE pg_catalog."default" NOT NULL,
9      "Branch Name" character varying COLLATE pg_catalog."default",
10     CONSTRAINT "Labor Account Details_pkey" PRIMARY KEY ("A/C No.", "IFSC Code")
11         INCLUDE("IFSC Code")
12 )
13
14 TABLESPACE pg_default;
15
16 ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Account Details"
17     OWNER to postgres;
18
19
20 SELECT * FROM "LaborList and Wages"."Labor Account Details"
```

Data output | Messages | Notifications

	A/C No. [PK] character varying	IFSC Code [PK] character varying	Branch Name character varying

Table: Labor Bank Details

Query | Query History

```
1  -- Table: LaborList and Wages.Labor Bank Details
2
3  -- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Bank Details";
4
5  CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Bank Details"
6  (
7      "UPI Id" character varying COLLATE pg_catalog."default" NOT NULL,
8      "Name" character varying COLLATE pg_catalog."default" NOT NULL,
9      "Labor Id" character varying COLLATE pg_catalog."default" NOT NULL,
10     "A/C No." character varying COLLATE pg_catalog."default" NOT NULL,
11     CONSTRAINT "Labor Bank Details_pkey" PRIMARY KEY ("UPI Id", "A/C No.")
12         INCLUDE("A/C No.")
13 )
14
15 TABLESPACE pg_default;
16
17 ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Bank Details"
18     OWNER to postgres;
19
20 SELECT * FROM "LaborList and Wages"."Labor Bank Details"
```

Data output | Messages | Notifications

	UPI Id [PK] character varying	Name character varying	Labor Id character varying	A/C No. [PK] character varying

Successfully run. Total query runtime: 85 mse

Table: Labor Details

Query | Query History

```sql
-- Table: LaborList and Wages.Labor Details

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Details";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Details"
(
    "Labor Id" character varying COLLATE pg_catalog."default" NOT NULL,
    "UPI Id" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Details_pkey" PRIMARY KEY ("Labor Id", "UPI Id")
        INCLUDE("Labor Id")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Details"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Labor Details"
```

Data output | Messages | Notifications

Labor Id [PK] character varying	UPI Id [PK] character varying

Table: Labor Division

Query | Query History

```sql
-- Table: LaborList and Wages.Labor Division

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Division";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Division"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Division " character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Division_pkey" PRIMARY KEY ("Labor ID", "Division ")
        INCLUDE("Division ")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Division"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Labor Division"
```

Data output | Messages | Notifications

Labor ID [PK] character varying	Division [PK] character varying

Table: Labor Email

Query | Query History

```sql
-- Table: LaborList and Wages.Labor Email

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Email";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Email"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Email" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Email_pkey" PRIMARY KEY ("Labor ID", "Email")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Email"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Labor Email"
```

Data output | Messages | Notifications

Labor ID [PK] character varying	Email [PK] character varying

Table: Labor Mobile No.

Query | Query History

```sql
-- Table: LaborList and Wages.Labor Mobile no.

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Mobile no.";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Mobile no."
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Mobile No." character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Mobile no._pkey" PRIMARY KEY ("Labor ID", "Mobile No.")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Mobile no."
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Labor Mobile no."
```

Data output | Messages | Notifications

Labor ID [PK] character varying	Mobile No. [PK] character varying

Total rows: 0 of 0 | Query complete 00:00:00.125 | Successfully run. Total query runtime: 1

Table: Labor Personal Details

```sql
-- Table: LaborList and Wages.Labor Personal Details

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Personal Details";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Personal Details"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Name" character varying COLLATE pg_catalog."default" NOT NULL,
    "Community " character varying COLLATE pg_catalog."default",
    "City ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Date of Birth" date
)

TABLESPACE pg_default;



ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Personal Details"
    OWNER to postgres;


SELECT * FROM "LaborList and Wages"."Labor Personal Details"
```

Data output | Messages | Notifications

Labor ID character varying	Name character varying	Community character varying	City ID character varying	Date of Birth date

Total rows: 60 of 60 | Query complete 00:00:00.148

Table: Labor Type of Work

Query | Query History

```sql
-- Table: LaborList and Wages.Labor Type of Work

-- DROP TABLE IF EXISTS "LaborList and Wages"."Labor Type of Work";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Labor Type of Work"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Type of Work" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Labor Type of Work_pkey" PRIMARY KEY ("Labor ID", "Type of Work")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Labor Type of Work"
    OWNER to postgres;


SELECT * FROM "LaborList and Wages"."Labor Type of Work"

```

Data output | Messages | Notifications

Labor ID [PK] character varying	Type of Work [PK] character varying

Table: LaborWage

```sql
-- Table: LaborList and Wages.LaborWage

-- DROP TABLE IF EXISTS "LaborList and Wages"."LaborWage";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."LaborWage"
(
    "Wage" character varying COLLATE pg_catalog."default" NOT NULL,
    "Wage Type" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "LaborWage_pkey" PRIMARY KEY ("Wage Type")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."LaborWage"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."LaborWage"
```

Data output | Messages | Notifications

Wage character varying	Wage Type [PK] character varying

Table: Payer Details

```sql
-- Table: LaborList and Wages.Payer Details

-- DROP TABLE IF EXISTS "LaborList and Wages"."Payer Details";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Payer Details"
(
    "UPI ID(Payer)" character varying COLLATE pg_catalog."default" NOT NULL,
    "Supervisor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Payer Details_pkey" PRIMARY KEY ("UPI ID(Payer)", "Supervisor ID")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Payer Details"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Payer Details"
```

Data output | Messages | Notifications

UPI ID(Payer) [PK] character varying	Supervisor ID [PK] character varying

Table: Schedule

Query | Query History

```sql
-- Table: LaborList and Wages.Schedule

-- DROP TABLE IF EXISTS "LaborList and Wages"."Schedule";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Schedule"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Date" date NOT NULL,
    "Entry time" time without time zone,
    "Exit  time" time without time zone,
    "Supervisor ID" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Schedule _pkey" PRIMARY KEY ("Labor ID", "Date")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Schedule"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Schedule"
```

Data output | Messages | Notifications

Labor ID [PK] character varying	Date [PK] date	Entry time time without time zone	Exit time time without time zone	Supervisor ID character varying

Successfully run. Total query runtime: 158 msec.

Total rows: 0 of 0 | Query complete 00:00:00.158

Table: Payment

```sql
-- Table: LaborList and Wages.Payment

-- DROP TABLE IF EXISTS "LaborList and Wages"."Payment";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Payment"
(
    "Labor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Date" character varying COLLATE pg_catalog."default" NOT NULL,
    "UPI ID (Reciver)" character varying COLLATE pg_catalog."default" NOT NULL,
    "Amount" bigint NOT NULL,
    "Payment ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Wage Type" character varying COLLATE pg_catalog."default",
    "UPI ID (Payer)" character varying COLLATE pg_catalog."default",
    CONSTRAINT "Payment_pkey" PRIMARY KEY ("Labor ID", "Date")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Payment"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Payment"
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Date [PK] character varying	UPI ID (Reciver) character varying	Amount bigint	Payment ID character varying	Wage Type character varying	UPI ID (Payer) character varying

Table: Supervisor Email

```sql
-- Table: LaborList and Wages.Supervisor Email

-- DROP TABLE IF EXISTS "LaborList and Wages"."Supervisor Email";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Supervisor Email"
(
    "Supervisor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Email" character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "Supervisor Email_pkey" PRIMARY KEY ("Supervisor ID", "Email")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Supervisor Email"
    OWNER to postgres;

SELECT * FROM "LaborList and Wages"."Supervisor Email"
```

Data output | Messages | Notifications

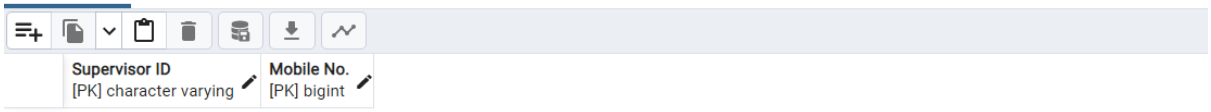
	Supervisor ID [PK] character varying	Email [PK] character varying

Table: Supervisor Mobile

Query | Query History

```sql
-- Table: LaborList and Wages.Supervisor Mobile

-- DROP TABLE IF EXISTS "LaborList and Wages"."Supervisor Mobile";

CREATE TABLE IF NOT EXISTS "LaborList and Wages"."Supervisor Mobile"
(
    "Supervisor ID" character varying COLLATE pg_catalog."default" NOT NULL,
    "Mobile No." bigint NOT NULL,
    CONSTRAINT "Supervisor Mobile_pkey" PRIMARY KEY ("Supervisor ID", "Mobile No.")
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS "LaborList and Wages"."Supervisor Mobile"
    OWNER to postgres;


SELECT * FROM "LaborList and Wages"."Supervisor Mobile"
```

Data output | Messages | Notifications



Supervisor ID [PK] character varying	Mobile No. [PK] bigint

Data Snapshot

Table: Supervisor

Query | Query History

```
1  COPY "LaborList and Wages"."Supervisor"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Supervisor.csv'
3  DELIMITER ',' CSV HEADER;
4
5  SELECT * FROM "LaborList and Wages"."Supervisor"
6
```

Loading...

Data output | Messages | Notifications

	Supervisor ID [PK] character varying	Rating double precision	No. of Laborers bigint
1	16-3034725	3	346
2	88-5202202	2	89
3	00-6811642	3.5	100
4	04-8870890	1.5	163
5	69-2899100	1.8	408
6	38-9326522	4.7	378
7	85-4216614	3.5	441
8	30-0027747	4.2	405
9	44-1768973	4.1	36
10	68-7550862	1.8	282

Total rows: 60 of 60 | Query complete 00:00:00.061

Table: City Details

```
1  COPY "LaborList and Wages"."City Details"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\City Details.csv'
3  DELIMITER ',' CSV HEADER;
4
5  SELECT * FROM "LaborList and Wages"."City Details"
6
```

Loading...

Data output | Messages | Notifications

	City ID [PK] character varying	Dept ID character varying	City Name character varying	No. of Laborers bigint	No. of Supervisors bigint
1	1	50-3092821	Klembivka	61	100
2	2	47-6417213	Daireaux	70	60
3	3	21-1308415	Šenov	73	15
4	4	76-1427397	Nanton	65	51
5	5	70-6189964	Xufu	57	14
6	6	65-3406190	Huangshi	22	28
7	7	44-0530295	Ngama	36	29
8	8	24-9635489	Pushkino	15	87
9	9	35-8406472	Mahaddayweyne	37	18
10	10	80-2507368	Falkenberg	60	

Successfully run. Total query

Total rows: 60 of 60 | Query complete 00:00:00.054

Table: Municipal Department

Query | Query History

```
1  COPY "LaborList and Wages"."Municipal Department"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Municipal Department.csv'
3  DELIMITER ',' CSV HEADER;
4
5  SELECT * FROM "LaborList and Wages"."Municipal Department"
6
```

Data output | Messages | Notifications

	Dept ID [PK] character varying	City ID character varying	Overall Rating double precision	Total Expense bigint	UPI ID(Payer) character varying
1	43-6334364	1	1.5	48066732745896	fusce
2	87-6353698	2	4.3	24840174584925	sapien
3	16-8612330	3	4.7	47000234538750	justo
4	52-5763403	4	3.6	39341917150661	eget
5	25-4247522	5	1.9	5422202944200	eget
6	35-8406472	6	2.9	53073595787280	mattis
7	08-4207697	7	4.1	20140052461463	quis
8	22-2505538	8	1.2	93126543565413	rutrum
9	66-7128972	9	3.2	29570910827469	ridiculus
10	45-7756531	10	2.2	2886133997124	ut

Successfully run. Total query runtime: 64

Total rows: 60 of 60 | Query complete 00:00:00.064

Table: Labor

Query | Query History

```
1  COPY "LaborList and Wages"."Labor"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor.csv'
3  DELIMITER ',' CSV HEADER;
4
5  SELECT * FROM "LaborList and Wages"."Labor"
6
```

Loading...

Data output | Messages | Notifications

	Labor ID [PK] character varying	Supervisor ID character varying	Rating double precision	City ID character varying	Name character varying	Gender character varying	Work character varying	Wage Type character varying
1	61-8325798	477-31-8522	4.1	59	Moina Shiels	Female	Craft and related ...	Weekly
2	19-8371386	333-69-9088	1	48	Heywood Guitton	Male	Clerks	Daily
3	10-3225613	667-75-6916	2.1	47	Georgine Menhci	Female	Craft and related ...	Weekly
4	45-8392744	503-02-8176	3	42	Brent Plackstone	Male	Service workers ...	Weekly
5	03-8098287	667-75-6916	3	58	Colas Crevagh	Male	Professionals	Daily
6	99-0564036	570-32-2796	3.7	50	Pearl Bulloch	Female	senior officials a...	Weekly
7	53-8469519	422-36-1326	4.2	40	Yurik Fetter	Male	Plant and machin...	Daily
8	41-3929175	705-30-2459	2.3	8	Lucia Ramage	Female	Elementary occu...	Daily
9	99-7848040	145-21-2384	1.5	10	Bartholomeo Gor...	Male	Skilled agricultur...	Daily
10	35-9076235	584-24-9297	2.9	34	Joletta Blaxter	Female	Skilled agricultur...	Daily

Total rows: 60 of 60 | Query complete 00:00:00.068 | Ln 6, Col 1

Table: Labor Account Details

Query | Query History

```
1  COPY "LaborList and Wages"."Labor Account Details"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Account Details.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Labor Account Details"
7
```

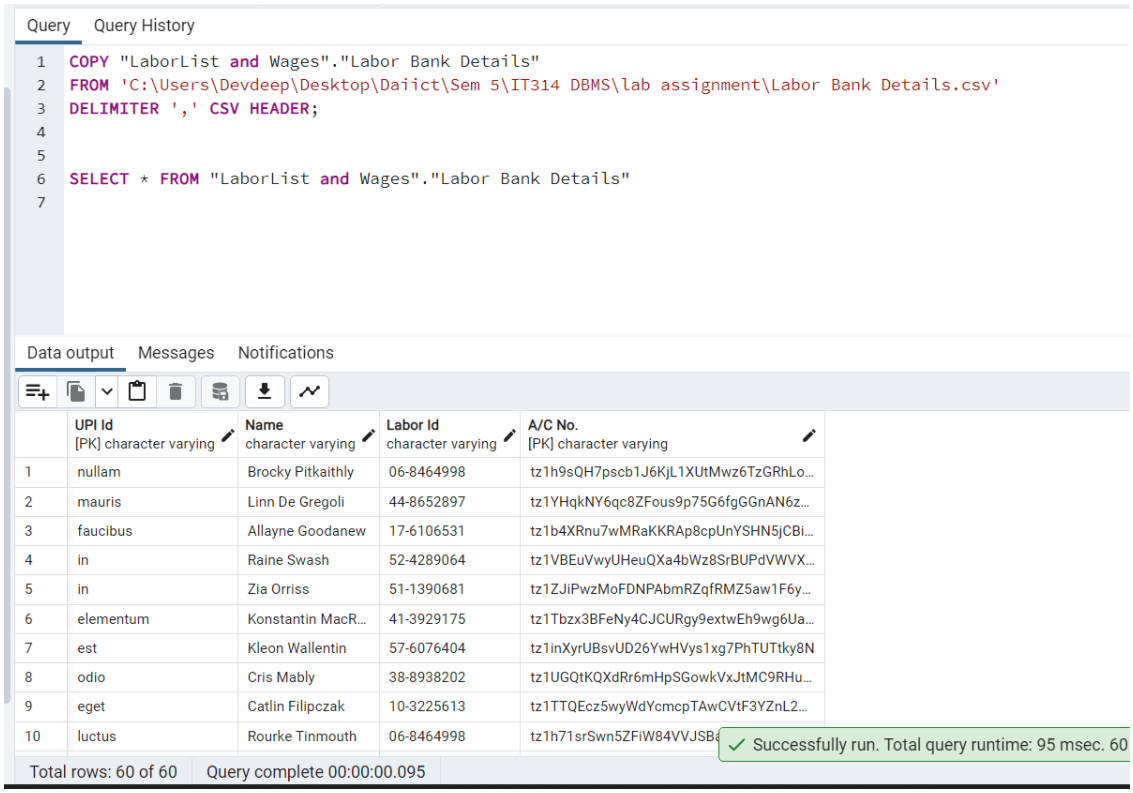
Data output | Messages | Notifications

	A/C No. [PK] character varying	IFSC Code [PK] character varying	Branch Name character varying
1	tz1Pz5x4334x7o5qw...	quam	Beo
2	tz1RKBXkrY4or3Qc8...	faucibus	Lau
3	tz1inXyrUBsvUD26Yw...	nulla	Balibago
4	tz1Pz5x4334x7o5qw...	ultrices	Palkovice
5	tz1Lit2NBxaas15UAF...	praesent	Thaba Nchu
6	tz1VhH6RAJEafTovLj...	ante	Raojān
7	tz1T7XUdir5pLGS6fA...	turpis	Đức Trọng
8	tz1gSFZSJkYji9u61p...	molestie	Savanna-la-Mar
9	tz1gN16neUJ4M8am...	vel	Badu
10	tz1QhCNrd71RDtq3rF...	sit	Xiaogang

Successfully run. Total query runtime: 59 msec. 60 rows a

Total rows: 60 of 60 | Query complete 00:00:00.059

Table: Labor Bank Details



Query | Query History

```
COPY "LaborList and Wages"."Labor Bank Details"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Bank Details.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."Labor Bank Details"
```

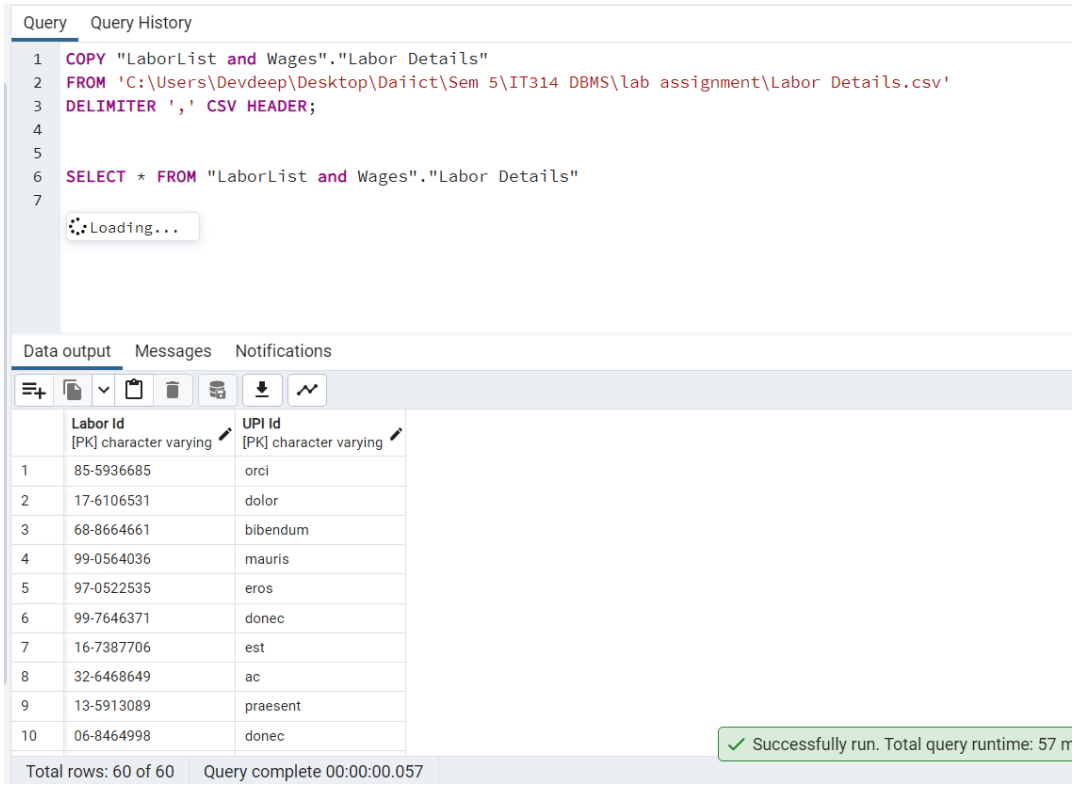
Data output | Messages | Notifications

	UPI Id [PK] character varying	Name character varying	Labor Id character varying	A/C No. [PK] character varying
1	nullam	Brocky Pitkaithly	06-8464998	tz1h9sQH7pscb1J6KjL1XUtMwz6TzGRhLo...
2	mauris	Linn De Gregoli	44-8652897	tz1YHqkNY6qc8ZFous9p75G6fgGGnAN6z...
3	faucibus	Allayne Goodanew	17-6106531	tz1b4XRnu7wMRaKKRAp8cpUnYSHN5jCBi...
4	in	Raine Swash	52-4289064	tz1VBEuVwyUHeuQXa4bWz8SrBUPdVWVX...
5	in	Zia Orriss	51-1390681	tz1ZJiPwzMoFDNPAbmRZqfRMZ5aw1F6y...
6	elementum	Konstantin MacR...	41-3929175	tz1Tbzx3BFeNy4CJCURgy9extwEh9wg6Ua...
7	est	Kleon Wallentin	57-6076404	tz1inXyrUBsvUD26YwHVys1xg7PhTUTtky8N
8	odio	Cris Mably	38-8938202	tz1UGQtKQXdRr6mHpSGowkVxJtMC9RHu...
9	eget	Catlin Filipczak	10-3225613	tz1TTQEcz5wyWdYcmcpTAwCVtF3YZnL2...
10	luctus	Rourke Tinmouth	06-8464998	tz1h71srSwn5ZFiW84VVJSB...

Successfully run. Total query runtime: 95 msec. 60

Total rows: 60 of 60 | Query complete 00:00:00.095

Table: Labor Details



Query | Query History

```
COPY "LaborList and Wages"."Labor Details"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Details.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."Labor Details"

Loading...
```

Data output | Messages | Notifications

	Labor Id [PK] character varying	UPI Id [PK] character varying
1	85-5936685	orci
2	17-6106531	dolor
3	68-8664661	bibendum
4	99-0564036	mauris
5	97-0522535	eros
6	99-7646371	donec
7	16-7387706	est
8	32-6468649	ac
9	13-5913089	praesent
10	06-8464998	donec

Successfully run. Total query runtime: 57 m

Total rows: 60 of 60 | Query complete 00:00:00.057

Table: Labor Division

Query | Query History

```
COPY "LaborList and Wages"."Labor Division"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Division.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."Labor Division"

```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Division [PK] character varying
1	73-1897695	Economic sector
2	67-8749143	Agriculture
3	36-8165380	Agriculture
4	89-9392887	Construction
5	84-8729652	Construction
6	75-6646197	Economic sector
7	97-0522535	Economic sector
8	16-4379098	Economic sector
9	75-6646197	Agriculture
10	99-7646371	Agriculture

Successfully run. Total quer

Total rows: 60 of 60 | Query complete 00:00:00.075

Table: Labor Email

Query | Query History

```
COPY "LaborList and Wages"."Labor Email"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Email.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."Labor Email"

```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Email [PK] character varying
1	73-1897695	lnovotni0@dagondesi...
2	89-9392887	ajanson1@archive.org
3	38-8938202	dadam2@1und1.de
4	19-8371386	kdevine3@mysql.com
5	19-8371386	sflecknell4@wordpre...
6	20-8916872	hcoryndon5@aol.com
7	03-8098287	dgoodlife6@noaa.gov
8	85-5936685	aarnholtz7@vinaora.c...
9	75-6646197	sflea8@yandex.ru
10	53-8469519	acarden9@globo.com

Successfully run. Total query runtime: 86

Total rows: 60 of 60 | Query complete 00:00:00.086

Table: Labor Mobile No.

Query | Query History

```
1  COPY "LaborList and Wages"."Labor Mobile no."
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Mobile no..csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Labor Mobile no."
7
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Mobile No. [PK] character varying
1	10-3225613	+389 537 478 1606
2	29-1869183	+86 526 245 3302
3	13-5913089	+86 856 432 6984
4	28-6908845	+48 485 359 9442
5	84-8729652	+7 579 918 9455
6	29-1869183	+63 871 867 3095
7	35-9076235	+63 925 418 5627
8	99-7646371	+86 833 666 5754
9	20-8916872	+54 646 925 3725
10	20-8916872	+351 938 540 9948

Total rows: 60 of 60 | Query complete 00:00:00.162

Table: Labor Personal Details

Query | Query History

```
1  COPY "LaborList and Wages"."Labor Personal Details"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Personal Details.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Labor Personal Details"
```

Loading...

Data output | Messages | Notifications

	Labor ID character varying	Name character varying	Community character varying	City ID character varying	Date of Birth date
1	68-8664661	Brigitta Reck	Urban	53	1971-03-29
2	16-7387706	Sharleen Ferrand	Urban	59	1971-01-06
3	96-5570833	Gypsy Lemin	Rural	15	1992-06-05
4	65-3192637	Flossi Dubble	Rural	50	2002-09-22
5	85-5936685	Darcee Norcliffe	Rural	42	1986-11-11
6	96-5570833	Mersey Lowdes	Rural	15	1973-09-01
7	57-6076404	Starr Trevains	Rural	51	1976-10-30

Total rows: 60 of 60 | Query complete 00:00:00.112

Table: Labor Type of Work

Query | Query History

```
1  COPY "LaborList and Wages"."Labor Type of Work"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Labor Type of Work.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Labor Type of Work"
7
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Type of Work [PK] character varying
1	19-8371386	Regular
2	76-1947531	Casual
3	27-7381874	Casual
4	26-7640693	Regular
5	17-6106531	Regular
6	96-5570833	Casual
7	87-1929125	Casual
8	35-9076235	Casual
9	05-3299496	Casual
10	97-0522535	Regular

Table: LaborWage

```
COPY "LaborList and Wages"."LaborWage"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\LaborWage.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."LaborWage"

```

Data output | Messages | Notifications

	Wage [PK] character varying	Wage Type [PK] character varying
1	2486	Daily
2	6376	Weekly
3	9313	Daily
4	2162	Weekly
5	1808	Weekly
6	5381	Weekly
7	6946	Weekly
8	5569	Daily
9	4679	Weekly
10	5980	Weekly

Total rows: 60 of 60 | Query complete 00:00:00.348

Table: Payer Details

Query | Query History

```
COPY "LaborList and Wages"."Payer Details"
FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Payer_detail.csv'
DELIMITER ',' CSV HEADER;


SELECT * FROM "LaborList and Wages"."Payer Details"

```

Data output | Messages | Notifications

	UPI ID(Payer) [PK] character varying	Supervisor ID [PK] character varying
1	erat	29-3454719
2	porttitor	61-6957948
3	duis	77-7624298
4	congue	75-0948013
5	nisi	34-6994225
6	maecenas	41-2547566
7	condimentum	46-1912510
8	fusce	98-5683490
9	vestibulum	09-3876245
10	magna	79-4011395

Successfully run. Total query runtir

Total rows: 60 of 60 | Query complete 00:00:00.707

Table: Schedule

Query | Query History

```
1  COPY "LaborList and Wages"."Schedule"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Schedule.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Schedule"
```

Data output | Messages | Notifications

	Labor ID character varying	Entry time time without time zone	Exit time time without time zone	Supervisor ID character varying	Date character varying
1	40-7333440	14:01:00	11:44:00	16-3034725	1/30/2022
2	73-1897695	18:56:00	18:01:00	88-5202202	12-03-2021
3	19-8371386	07:21:00	05:28:00	00-6811642	05-12-2022
4	05-3299496	17:07:00	06:43:00	04-8870890	2/18/2022
5	96-5570833	12:00:00	10:13:00	69-2899100	3/23/2022
6	36-8165380	19:57:00	18:15:00	38-9326522	02-05-2022
7	41-3929175	20:51:00	17:45:00	85-4216614	

Successfully run. Total query runtime: 102 ms

Total rows: 60 of 60 | Query complete 00:00:00.102

Table: Payment

Query | Query History

```
1  COPY "LaborList and Wages"."Payment"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Payment.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Payment"
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Date [PK] character varying	UPI ID (Reciver) character varying	Amount bigint	Payment ID character varying	Wage Type character varying	UPI ID (Payer) character varying
1	65-3192637	23/4/2022	faucibus	1847002	61-8099867	Weekly	vehicula
2	99-7848040	10/11/2022	donec	7327793	86-8633931	Daily	sem
3	76-1947531	29/1/2022	eros	1967573	15-3534436	Daily	mauris
4	29-1869183	25/9/2022	congue	3535521	48-0770407	Weekly	arcu
5	26-7640693	2/11/2022	quis	6095152	54-1467303	Weekly	ante
6	87-1929125	4/6/2022	risus	7431638	88-3822252	Daily	consequat
7	17-6106531	11/7/2022	quis	4647813	20-1606312		

Successfully run. Total query runtime: 175 msec.

Total rows: 60 of 60 | Query complete 00:00:00.175

Table: Supervisor Mobile

```
1  COPY "LaborList and Wages"."Supervisor Mobile"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Supervisor_mobile.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Supervisor Mobile"
```

Data output | Messages | Notifications

	Supervisor ID [PK] character varying	Mobile No. [PK] bigint
1	85-3168930	7428904110
2	08-7814367	899503225
3	61-0810428	5034163853
4	46-1912510	8504086828
5	17-9868882	8473791584
6	79-4011395	805984194
7	95-5744420	6628031594

Successfully run. Total query runtim

Total rows: 60 of 60 | Query complete 00:00:00.159

Table: Supervisor Email

Query | Query History

```
1  COPY "LaborList and Wages"."Supervisor Email"
2  FROM 'C:\Users\Devdeep\Desktop\Daiict\Sem 5\IT314 DBMS\lab assignment\Supervisor_email.csv'
3  DELIMITER ',' CSV HEADER;
4
5
6  SELECT * FROM "LaborList and Wages"."Supervisor Email"
```

Data output | Messages | Notifications

	Supervisor ID [PK] character varying	Email [PK] character varying
1	55-5119763	csellner0@about.com
2	54-5097200	mphifer1@ocn.ne.jp
3	43-8212884	jhellwing2@mayoclini...
4	29-4301702	tpavier3@addtoany.c...
5	52-5101189	mstothart4@sphinn.c...
6	29-4301702	mvala5@indiatimes.c...
7	43-8212884	sphette6@canalblog....

Successfully run. Total query runt

Total rows: 60 of 60 | Query complete 00:00:00.106

Queries:

1. Calculate Age from Date of Birth.

```
1
2
3  ALTER TABLE "LaborList and Wages"."Labor Personal Details"
4  ADD "AGE" INT;
5
```

```
ALTER TABLE

Query returned successfully in 158 msec.
```

2. Adding a New column Using Update and Set into Table

```
6  Update "LaborList and Wages"."Labor Personal Details"
7  SET "Age"=EXTRACT(YEAR FROM AGE("Labor Personal Details"."Date of Birth"))
8  SELECT * FROM "LaborList and Wages"."Labor Personal Details"
```

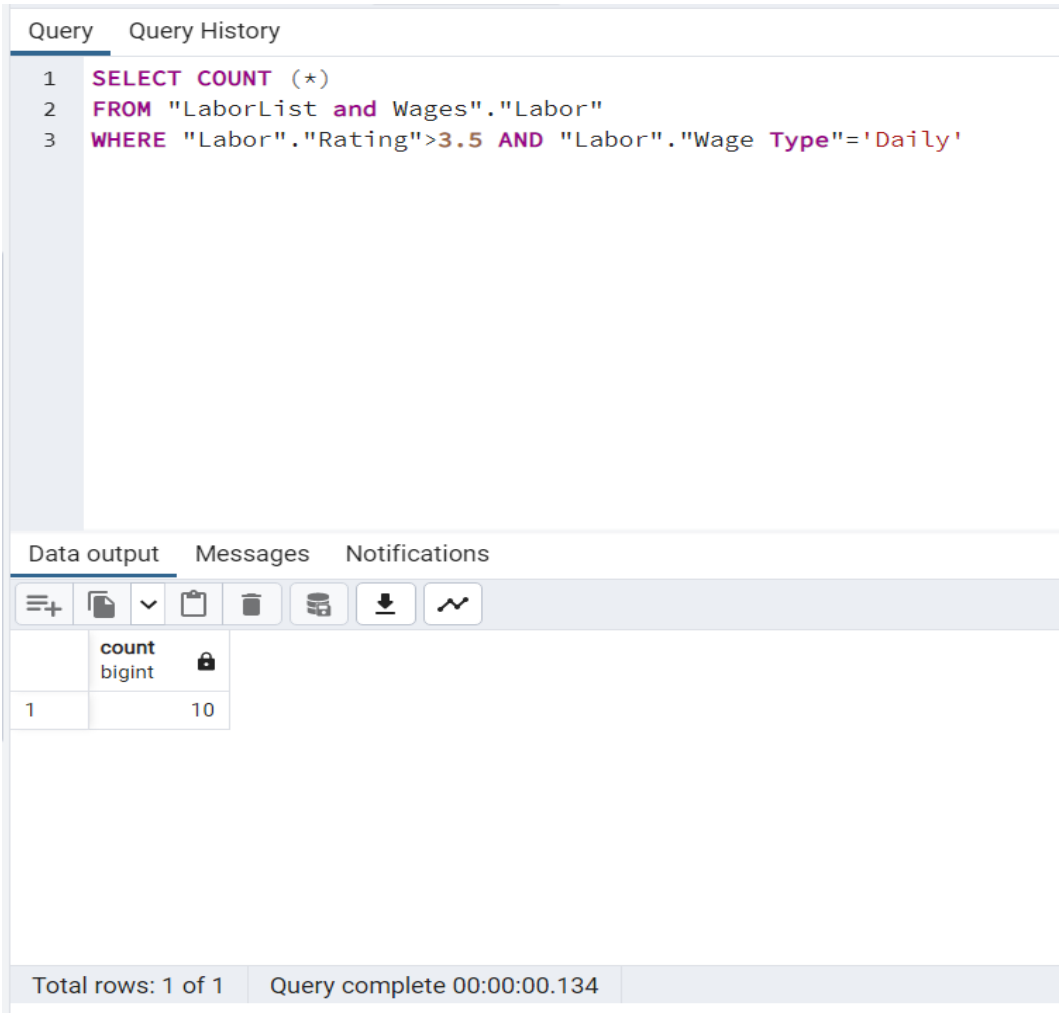
Data output | Messages | Notifications

	Labor ID character varying	Name character varying	Community character varying	City ID character varying	Date of Birth date	Age integer
1	40-7333440	Louie Domenicone	Rural	60	1989-01-06	33
2	09-6594491	Clotilda Dawid	Urban	10	1994-03-11	28
3	16-4379098	Christi Farren	Urban	31	1984-11-18	37
4	06-8464998	Ginny Huitson	Rural	50	1984-02-05	38
5	68-8664661	Brigitta Reck	Urban	53	1971-03-29	51
6	16-7387706	Sharleen Ferrand	Urban	59	1971-01-06	51
7	96-5570833	Gypsy Lemin	Rural	15	1992-06-05	30
8	65-3192637	Flossi Dubble	Rural	50	2002-09-2	

Successfully run. To

Total rows: 60 of 60 | Query complete 00:00:00.113

3. Count the Daily wagers having a rating greater than 3.5



4. Payments made on 27th November 2021

```
SELECT *
FROM "LaborList and Wages"."Payment"
WHERE "Payment"."Date"='27/11/2021'
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Date [PK] character varying	UPI ID (Reciver) character varying	Amount bigint	Payment ID character varying	Wage Type character varying	UPI ID (Payer) character varying
1	38-8938202	27/11/2021	amet	963551	09-2069538	Daily	sem

5. Find out the Unique type of works in the Database from Labor

Query | Query History

```
SELECT "Labor"."Work"
FROM "LaborList and Wages"."Labor"
GROUP BY "Labor"."Work"
```

Data output | Messages | Notifications

	Work character varying
1	Professionals
2	Legislators
3	senior officials and managers
4	Technicians and associate professionals
5	Elementary occupations
6	Service workers and shop and market sales work...
7	Skilled agricultural and fishery workers

Total rows: 10 of 10 | Query complete 00:00:00.148

6. Number of Municipal Department Having total expense greater than Avg Expense

Query | Query History

```
1
2  -- Number of Municipal Department Having total expense greater than Avg Expense
3  SELECT COUNT (*)
4  FROM "LaborList and Wages"."Municipal Department"
5  WHERE ("Total Expense" >
6         (SELECT AVG("Total Expense") FROM "LaborList and Wages"."Municipal Department")
7        )
```

Data output | Messages | Notifications

	count bigint
1	30

7. Number of Laborers having Age greater than 35.

Query | Query History

```
1  -- Number of Laborers having Age greater than 35
2  SELECT COUNT (*)
3  FROM "LaborList and Wages"."Labor Personal Details"
4  WHERE "Labor Personal Details"."Age">35
```

Data output | Messages | Notifications

	count bigint
1	39

Total rows: 1 of 1 | Query complete 00:00:36.035

8. Mobile Number of Laborers having Age greater than 23.

Query | Query History

```
1  -- Mobile Number of Laborers having Age 23
2  SELECT "Mobile No."
3  FROM "LaborList and Wages"."Labor Personal Details"
4    FULL OUTER JOIN "LaborList and Wages"."Labor Mobile no."
5    ON "Labor Mobile no."."Labor ID" = "Labor Personal Details"."Labor ID"
6  WHERE "Labor Personal Details"."Age" > 23
7
8
```

Data output | Messages | Notifications

	Mobile No. character varying
1	+256 624 470 65...
2	+63 450 335 4036
3	+7 468 769 0728
4	+86 781 687 4726
5	+7 640 874 6326
6	+7 765 646 2863
7	+48 259 365 8976

Total rows: 70 of 70 | Query complete 00:00:00.077 | Success

9. List of Top 5 cities having the highest expense

Query | Query History

```
1  -- List of Top 5 city having higest expense
2  SELECT *
3  FROM "LaborList and Wages"."Municipal Department"
4  ORDER BY "Total Expense"
5  DESC LIMIT 5
6
7
```

Data output | Messages | Notifications

	Dept ID [PK] character varying	City ID character varying	Overall Rating double precision	Total Expense bigint	UPI ID(Payer) character varying
1	34-6536856	29	3.3	99875326832116	eget
2	21-1308415	21	1.5	99810911190286	erat
3	72-7624843	27	2.3	95389711331180	ridiculus
4	65-3406190	48	2.4	94667536981599	vestibulum
5	22-2505538	8	1.2	93126543565413	rutrum

Total rows: 5 of 5 | Query complete 00:00:00.068

10. Find the contact details of Supervisors having No. of Laborers Greater than 10 using view:

Query | Query History

```
1  -- Find the contact details of Supervisors having No. of Laborers Greater than 10
2  CREATE OR REPLACE VIEW "Supervisor Info" AS
3  SELECT "Supervisor Mobile"."Mobile No."
4  FROM "LaborList and Wages"."Supervisor"
5  FULL OUTER JOIN "LaborList and Wages"."Supervisor Mobile"
6  ON "Supervisor Mobile"."Supervisor ID" = "Supervisor"."Supervisor ID" ;
7
8  SELECT * FROM "Supervisor Info" WHERE "Mobile No." IS NOT NULL
9
```

Loading...

Data output | Messages | Notifications

	Mobile No. bigint
1	7428904110
2	899503225
3	5034163853
4	8504086828
5	8473791584
6	805984194
7	6628031594

Total rows: 60 of 60 | Query complete 00:00:00.071

11. Use of Trigger:

Query | Query History

```
1  CREATE OR REPLACE FUNCTION func() RETURNS TRIGGER AS
2  $devansh_bhai_op$
3  BEGIN
4  IF new."Labor Id" IN
5  (SELECT "Labor Id" FROM "LaborList and Wages"."Labor Details")
6  THEN
7  RAISE EXCEPTION 'Sorry Bro! This value of Primary key ID already exists.';
8  ELSE
9  RAISE EXCEPTION 'Sorry Bro! This value of Primary key ID does not exists.';
10 END IF;
11 RETURN NEW;
12 END;
13 $devansh_bhai_op$
14 LANGUAGE plpgsql;
15
```

```
17
18  CREATE OR REPLACE TRIGGER devdeep
19  BEFORE INSERT ON  "LaborList and Wages"."Labor Details"
20  FOR EACH ROW EXECUTE PROCEDURE func();
21
22
23  INSERT INTO "LaborList and Wages"."Labor Details" VALUES (12234,'sldfjdfisj')
24
```

Loading...

Data output | Messages | Notifications

```
ERROR:  Sorry Bro! This value of Primary key ID does not exists.
CONTEXT:  PL/pgSQL function func() line 8 at RAISE
SQL state: P0001
```

12. Trigger (same as above but for a different case)

```
1   CREATE OR REPLACE FUNCTION func() RETURNS TRIGGER AS
2   $devansh_bhai_op$
3   BEGIN
4   IF new."Labor Id" IN
5   (SELECT "Labor Id" FROM "LaborList and Wages"."Labor Details")
6   THEN
7   RAISE EXCEPTION 'Sorry Bro! This value of Primary key ID already exists.';
8   ELSE
9   RAISE EXCEPTION 'Sorry Bro! This value of Primary key ID does not exists.';
10  END IF;
11  RETURN NEW;
12  END;
13  $devansh_bhai_op$
14  LANGUAGE plpgsql;
15
16
17
18  CREATE OR REPLACE TRIGGER devdeep
19  BEFORE INSERT ON  "LaborList and Wages"."Labor Details"
20  FOR EACH ROW EXECUTE PROCEDURE func();
21
22
23  INSERT INTO "LaborList and Wages"."Labor Details" VALUES ('85-5936685','jjjjjjjaaaannnnuuuu')
24
```

Data output | Messages | Notifications

```
ERROR:  Sorry Bro! This value of Primary key ID already exists.
CONTEXT:  PL/pgSQL function func() line 6 at RAISE
SQL state: P0001
```

13. Function to calculate the maximum amount of transaction between the Labor and Municipal Department :

```
CREATE OR REPLACE FUNCTION max_amount_transaction()
RETURNS TABLE (Amount BIGINT) AS
$$
BEGIN
RETURN QUERY SELECT MAX("Payment"."Amount") FROM "LaborList and Wages"."Payment";
END;
$$ LANGUAGE plpgsql;

SELECT * FROM max_amount_transaction();
```

Data output | Messages | Notifications

	amount bigint
1	9929679

14. 5 Supervisors having Lowest Laborers

```
-- 5 Supervisors having Lowest Laborers
SELECT *
FROM "LaborList and Wages"."Supervisor"
ORDER BY "No. of Laborers"
ASC LIMIT 5
```

Loading...

Data output | Messages | Notifications

	Supervisor ID [PK] character varying	Rating double precision	No. of Laborers bigint
1	19-3559675	4.7	10
2	22-4359226	1.2	12
3	44-1768973	4.1	36
4	49-7319336	3.5	39
5	68-6372392	2.8	57

Total rows: 5 of 5 | Query complete 00:00:00.098

15. Municipality Department having minimum rating

```
-- Municipality Department having minimum rating
SELECT * FROM "LaborList and Wages"."Municipal Department"
WHERE "Municipal Department"."Overall Rating"=
       (SELECT MIN("Municipal Department"."Overall Rating")
        FROM "LaborList and Wages"."Municipal Department")
```

Data output | Messages | Notifications

	Dept ID [PK] character varying	City ID character varying	Overall Rating double precision	Total Expense bigint	UPI ID(Payer) character varying
1	22-0279489	12	1	58173128240367	nulla

16. List of all Female Laborers

```
-- List of all Female Laborers
SELECT *
FROM "LaborList and Wages"."Labor"
WHERE "Labor"."Gender"='Female'
```

Data output | Messages | Notifications

	Labor ID [PK] character varying	Supervisor ID character varying	Rating double precision	City ID character varying	Name character varying	Gender character varying	Work character varying	Wage Type character varying
1	61-8325798	477-31-8522	4.1	59	Moina Shiels	Female	Craft and related ...	Weekly
2	10-3225613	667-75-6916	2.1	47	Georgine Menhci	Female	Craft and related ...	Weekly
3	99-0564036	570-32-2796	3.7	50	Pearl Bulloch	Female	senior officials a...	Weekly
4	41-3929175	705-30-2459	2.3	8	Lucia Ramage	Female	Elementary occu...	Daily
5	35-9076235	584-24-9297	2.9	34	Joletta Blaxter	Female	Skilled agricultur...	Daily
6	17-6106531	667-75-6916	1.6	24	Agna Hoofe	Female	Service workers ...	Weekly
7	97-0522535	825-78-7133	4.8	50	Yettie Reihm	Female	Elementary occu...	Daily
8	84-8729652	623-30-4646	3.7	49	Ines Jacquot	Female	Plant and machin...	Daily
9	40-7333440	422-36-1326	2.5	60	Ida Meegin	Female	Professionals	Weekly
10	73-1897695	407-64-2415	4	60	Nerita Slemmonds	Female	Legislators	Weekly
11	48-2440117	113-36-0963	3.6	24	Jenifer McGrory	Female	Technicians and ...	Weekly

Total rows: 32 of 32 | Query complete 00:00:00.155 | Ln 4, Col 2

17. List of Laborers worked on the 11th December 2022

```
1  -- List of Laborers worked on the 11th December 2022
2  SELECT *
3  FROM "LaborList and Wages"."Schedule"
4  WHERE "Schedule"."Date"='11-12-2022'
```

Data output | Messages | Notifications

	Labor ID character varying	Entry time time without time zone	Exit time time without time zone	Supervisor ID character varying	Date character varying
1	65-3192637	19:22:00	04:24:00	90-9369661	11-12-2022

18.
Understanding the Type of Work of Divisions using Frequency

```
1  -- Understanding Type of Work of Divisons using Frequency
2  SELECT "Labor Type of Work"."Type of Work" , COUNT (*)
3  FROM "LaborList and Wages"."Labor Type of Work"
4  GROUP BY "Labor Type of Work"."Type of Work"
5
6
```

Loading...

Data output | Messages | Notifications

	Type of Work character varying	count bigint
1	Regular	33
2	Casual	27

19. Calculating Average Laborers in City

```
1  CREATE OR REPLACE VIEW "MY view" AS
2  SELECT AVG("No. of Laborers ")
3  FROM "LaborList and Wages"."City Details";
4
5  SELECT * FROM "MY view"
6
7
8
```

Data output | Messages | Notifications

	avg numeric
1	52.76666666

20. Top 20 Cities with Expenses Greater than 1 million.

```
1  SELECT *
2  FROM "LaborList and Wages"."Municipal Department"
3  WHERE "Municipal Department"."Total Expense" > 1000000
4  ORDER BY "Municipal Department"."Total Expense"
5  DESC LIMIT 20
6
7
8  Loading...
```

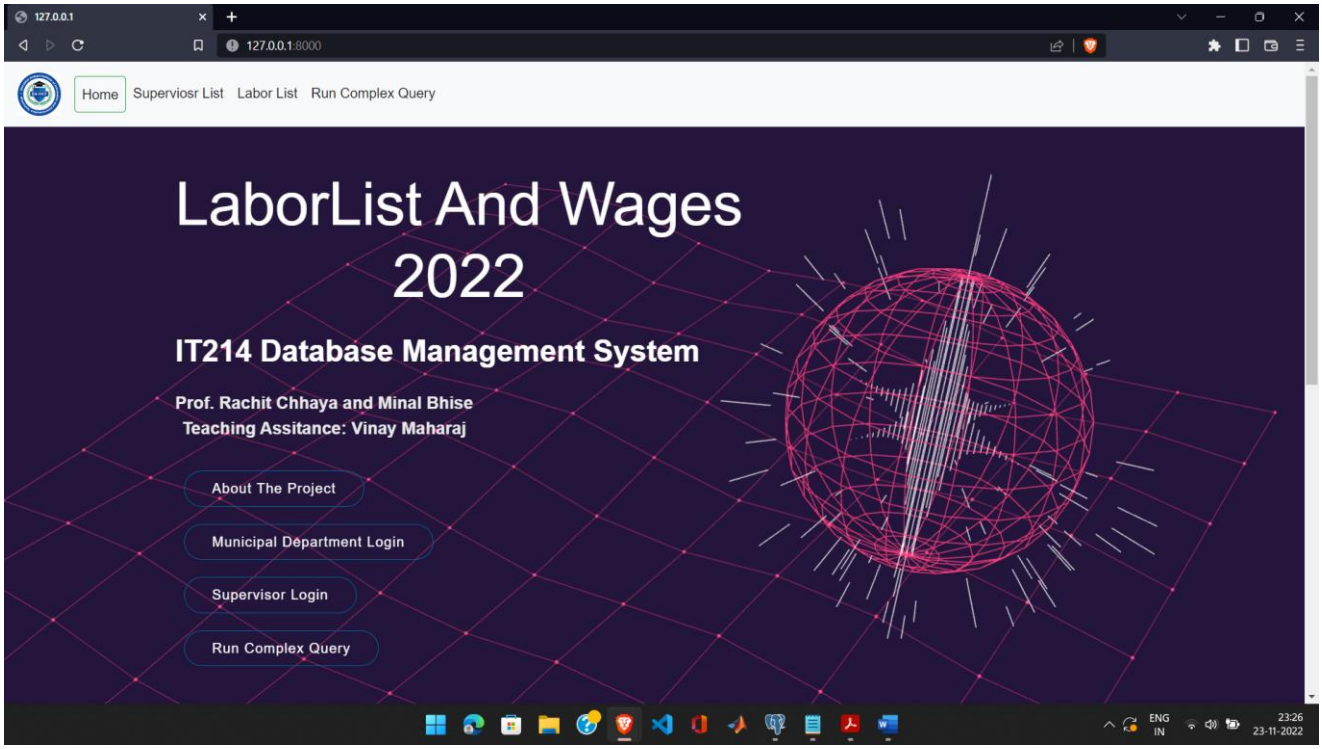
Data output | Messages | Notifications

	Dept ID [PK] character varying	City ID character varying	Overall Rating double precision	Total Expense bigint	UPI ID(Payer) character varying
1	34-6536856	29	3.3	99875326832116	eget
2	21-1308415	21	1.5	99810911190286	erat
3	72-7624843	27	2.3	95389711331180	ridiculus
4	65-3406190	48	2.4	94667536981599	vestibulum
5	22-2505538	8	1.2	93126543565413	rutrum
6	45-2243171	60	2.5	91225656061574	nibh
7	35-7881784	11	3.7	90939305051792	eros
8	41-9400167	56	4.9	87982419406480	erat
9	44-0530295	40	2.9	87250816854990	morbi
10	50-3092821	26	2.9	85104485242987	pretium
11	08-2772989	47	1.6	69587655613997	eu

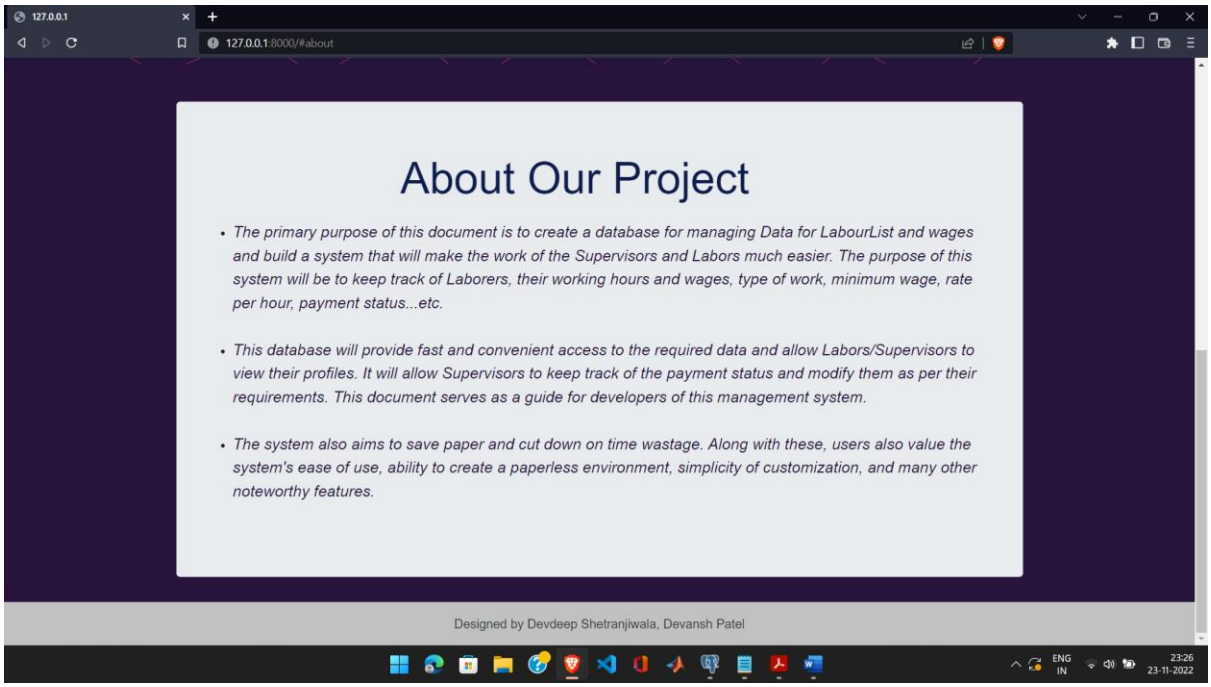
Total rows: 20 of 20 | Query complete 00:00:00.731

Screenshot of Demo interface/Website for the project :

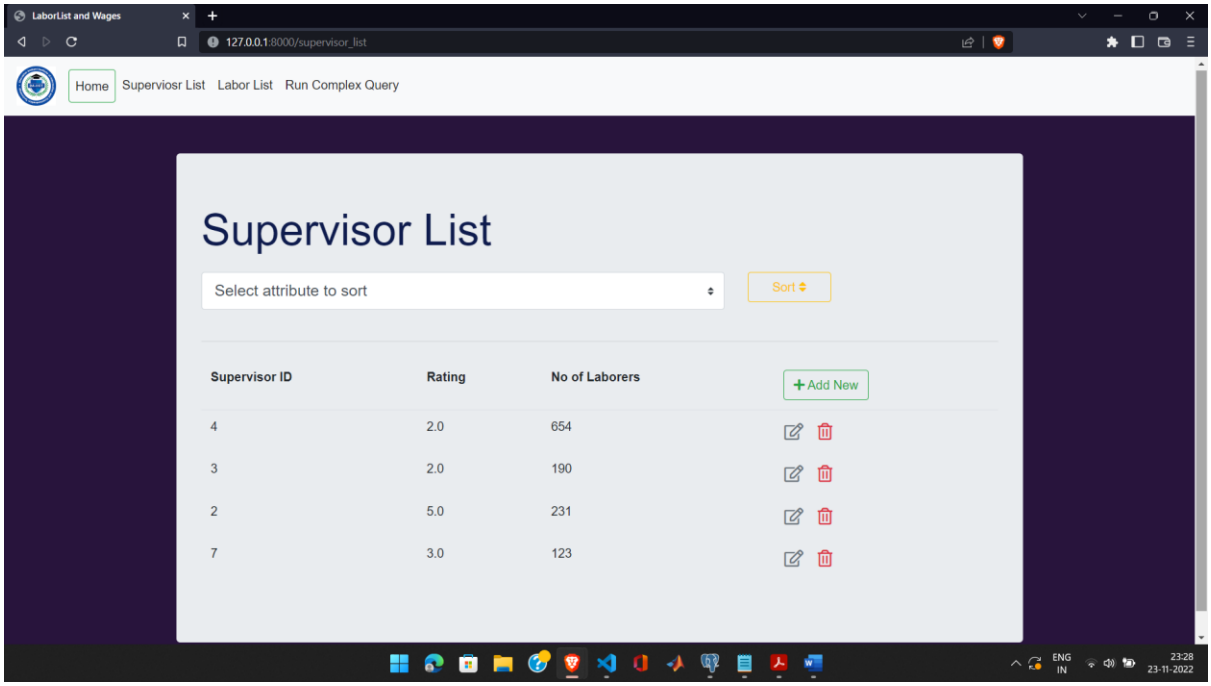
1. Home page :



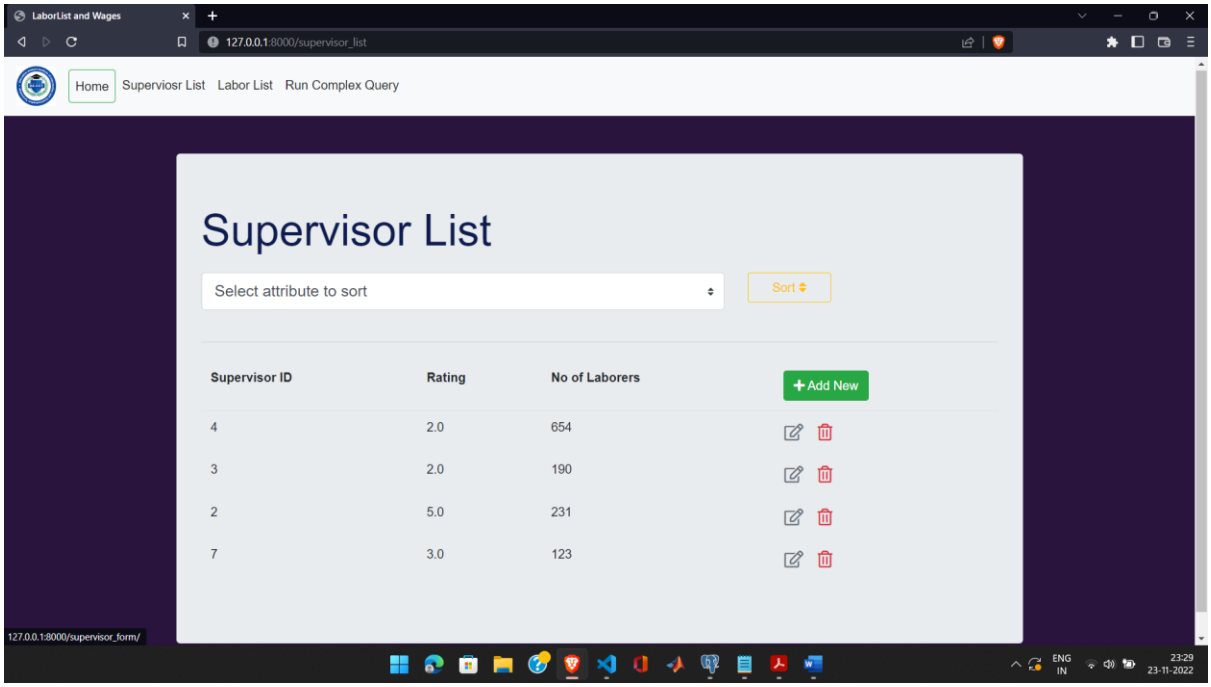
2. About Section from the button provided in home page:

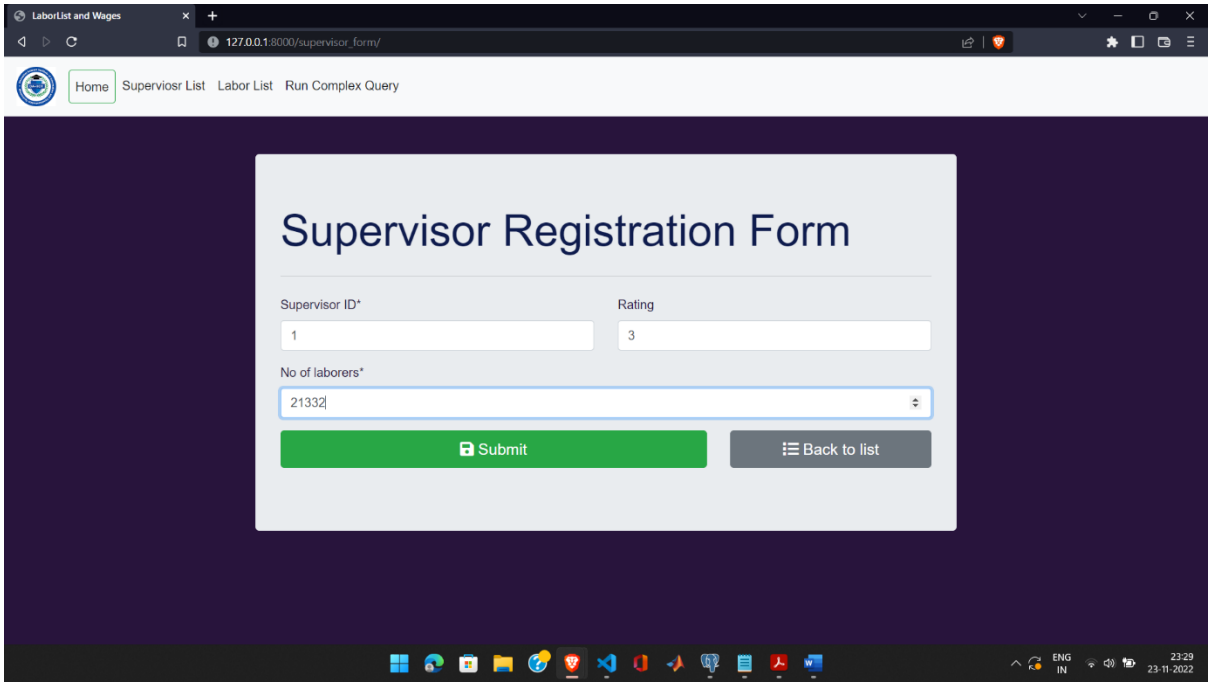


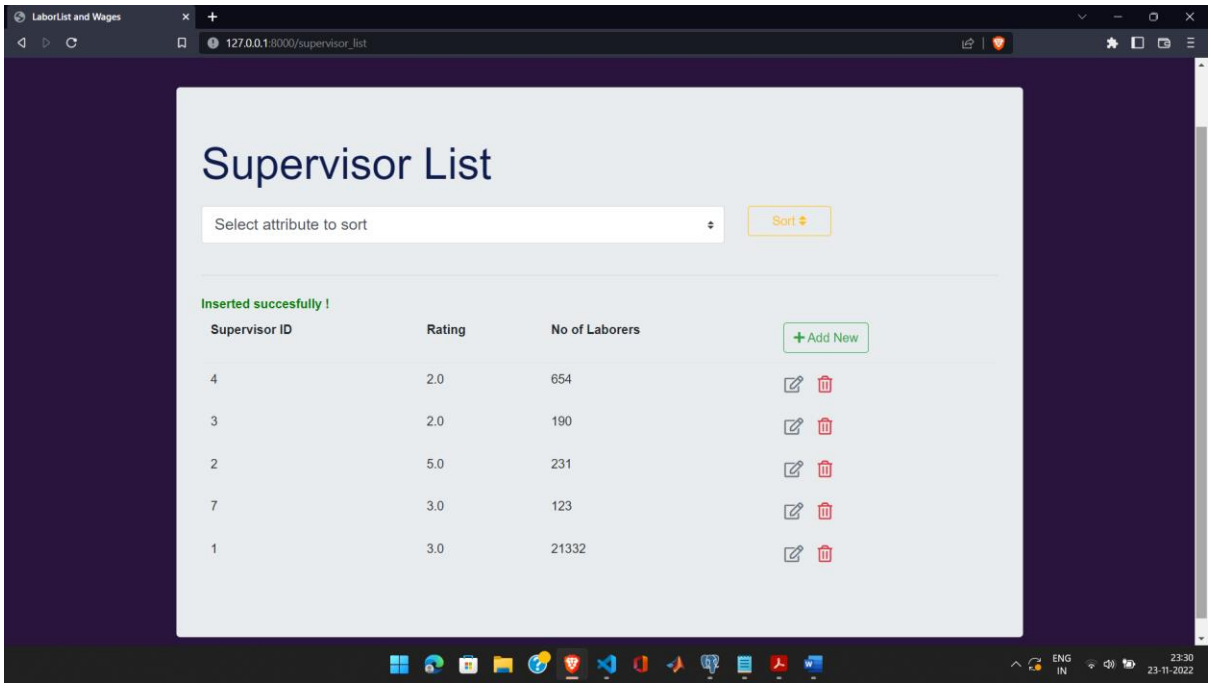
3. Municipal Login/Supervisor List :



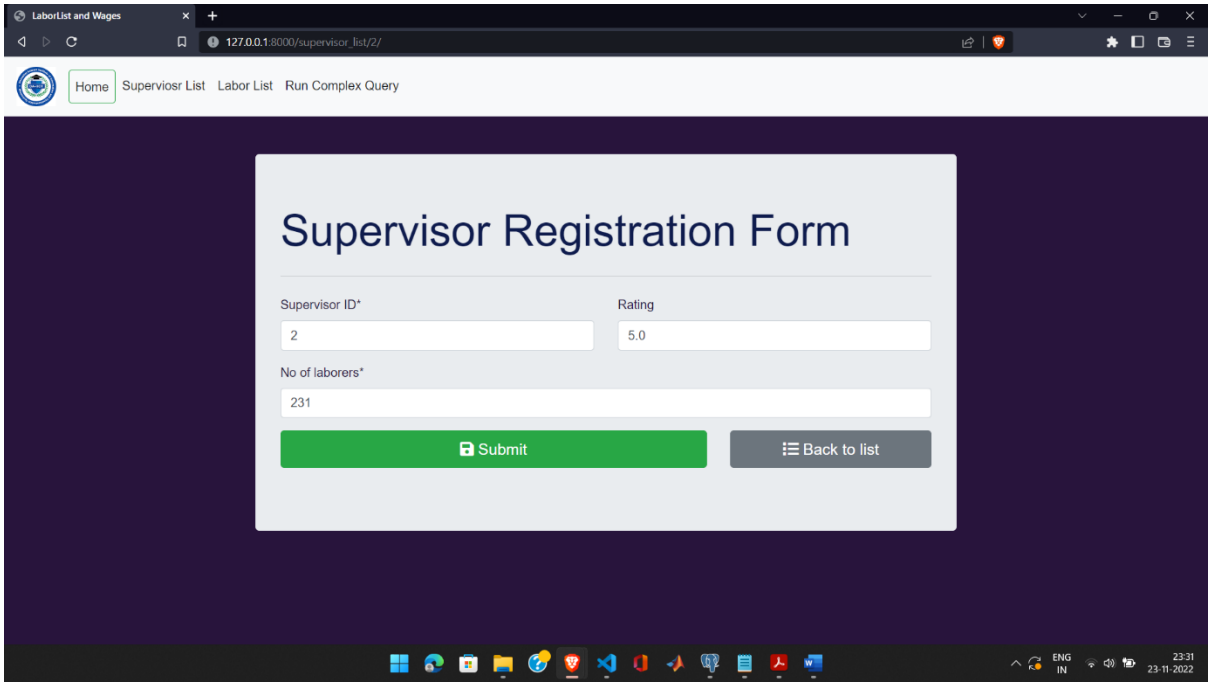
4. Insert Supervisor using Add new button:

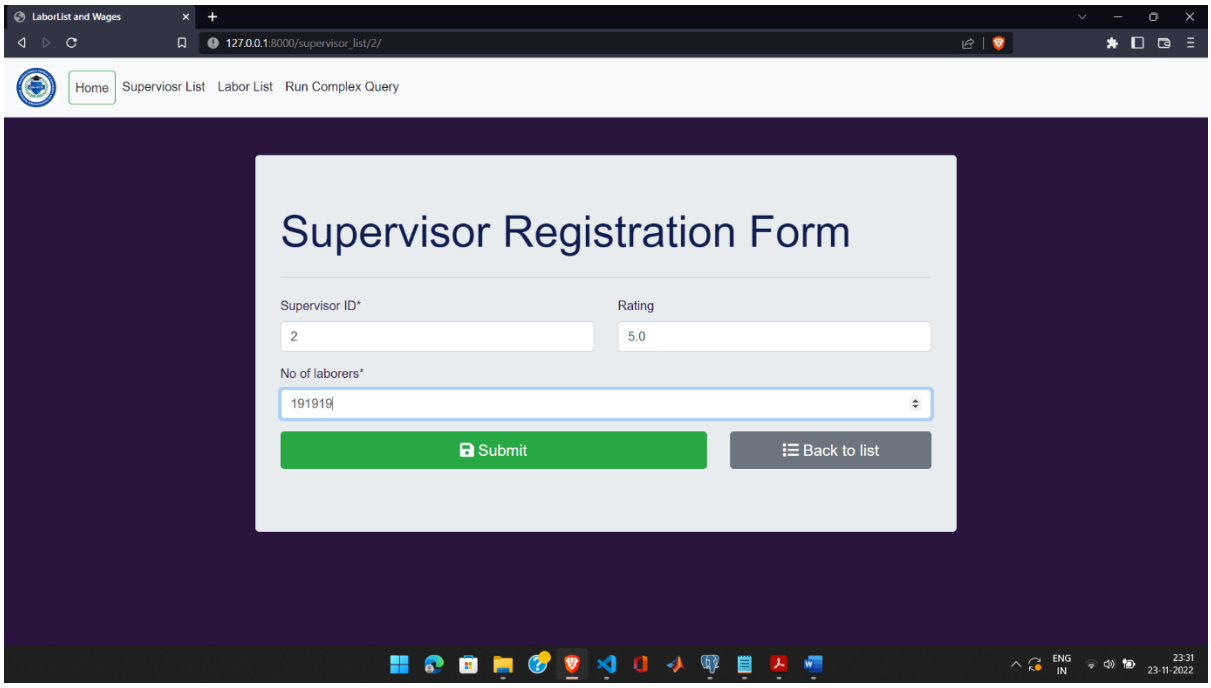


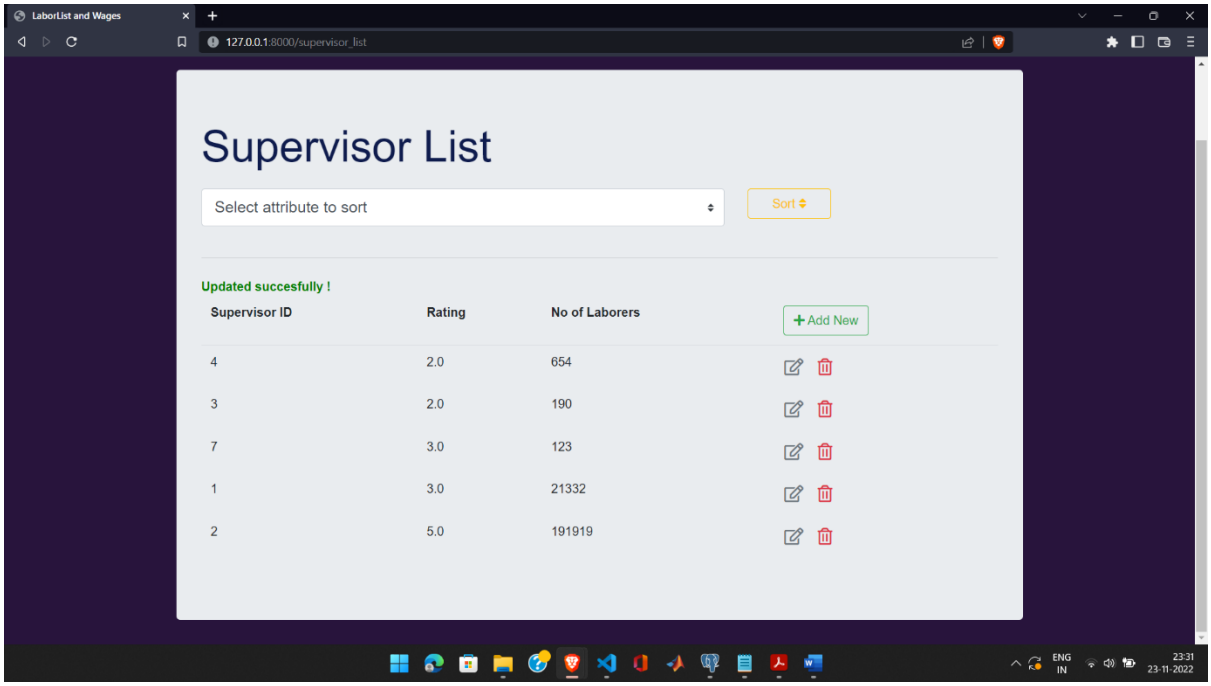




5. Update Supervisor details using the update button :

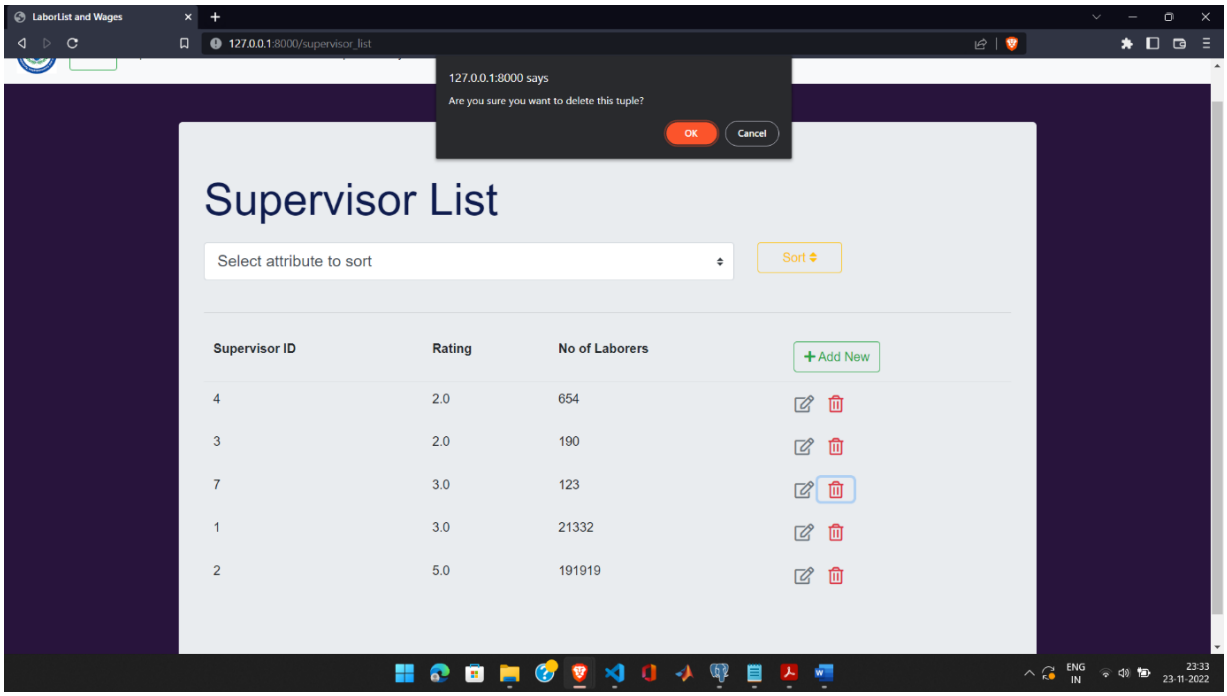

LaborList and Wages
127.0.0.1:8000/supervisor_list/2/
Home
Supervisor List
Labor List
Run Complex Query
Supervisor Registration Form
Supervisor ID*
2
Rating
5.0
No of laborers*
231
Submit
Back to list
ENG
IN
23:31
23-11-2022

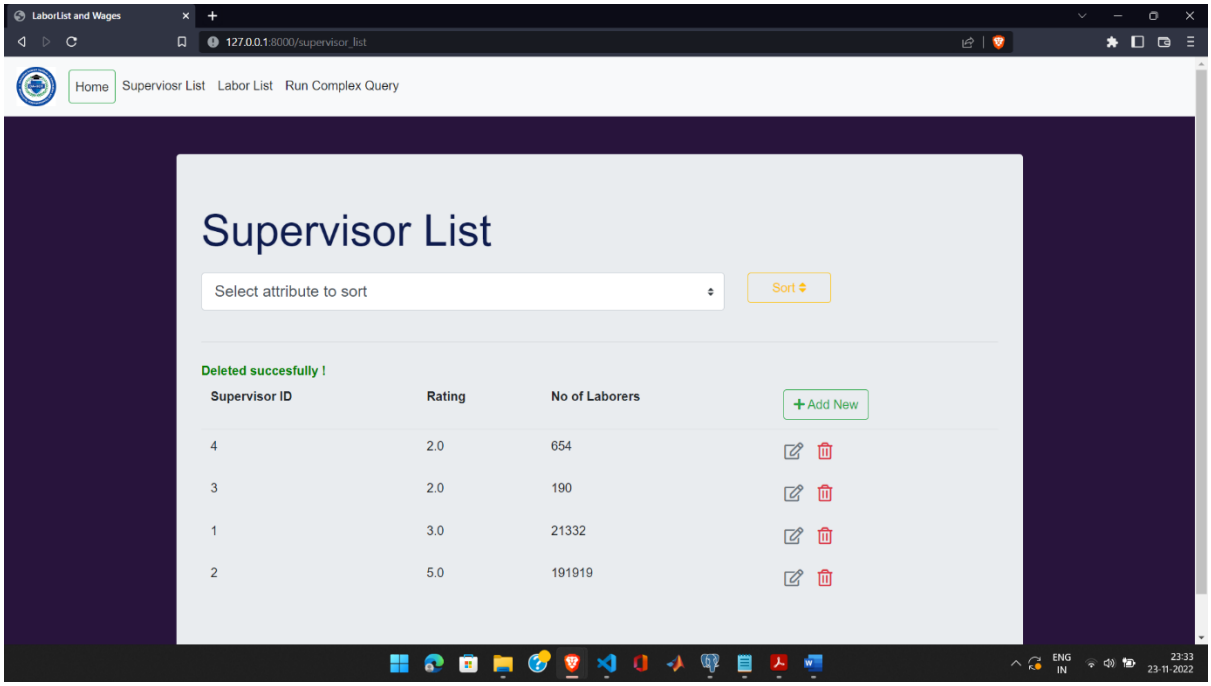

LaborList and Wages
127.0.0.1:8000/supervisor_list/2/
Home
Supervisor List
Labor List
Run Complex Query
Supervisor Registration Form
Supervisor ID*
2
Rating
5.0
No of laborers*
191919
Submit
Back to list
ENG
IN
23:31
23-11-2022



6. Delete Supervisor with trash button

(Delete will ask for conformation before deleting the data)

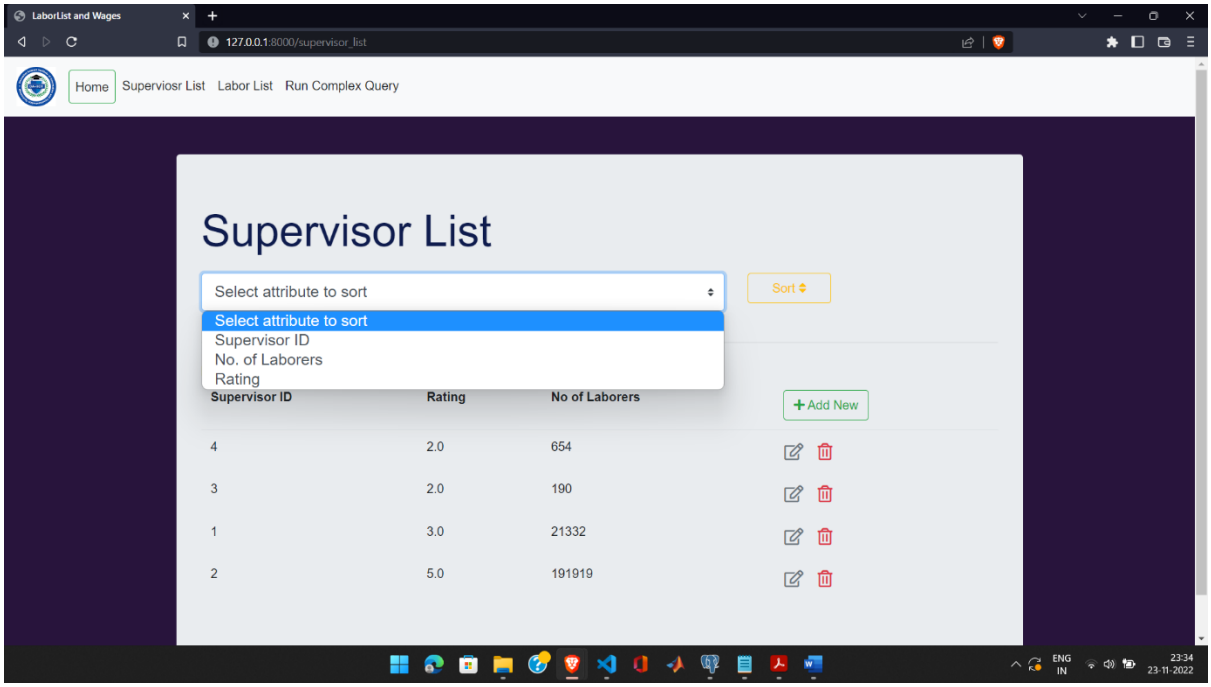


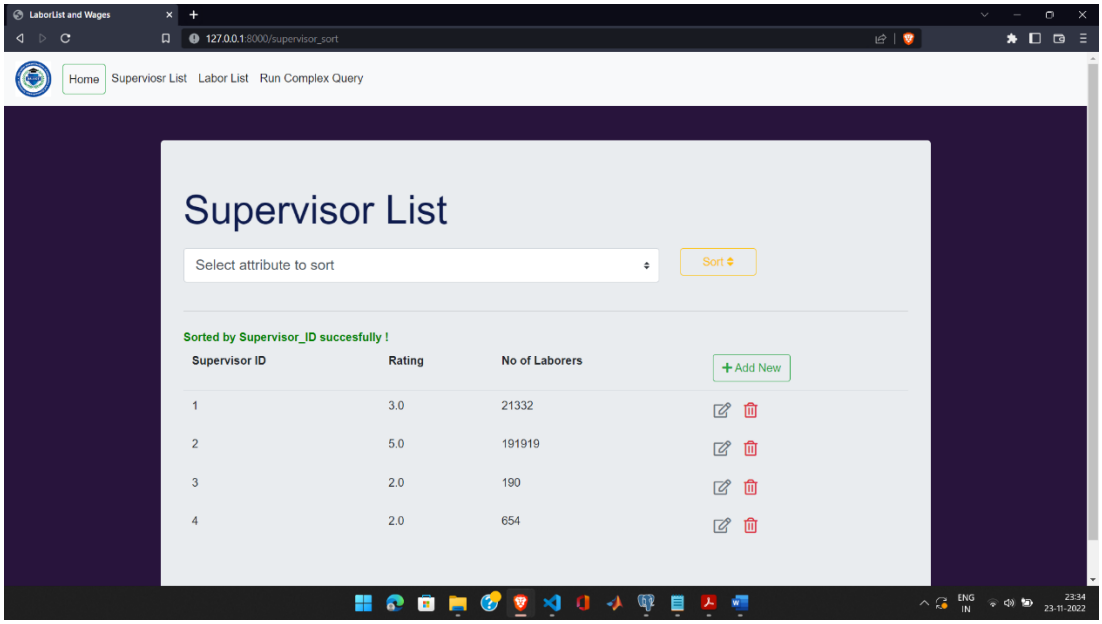


7. Sort function (Supervisor Data) :

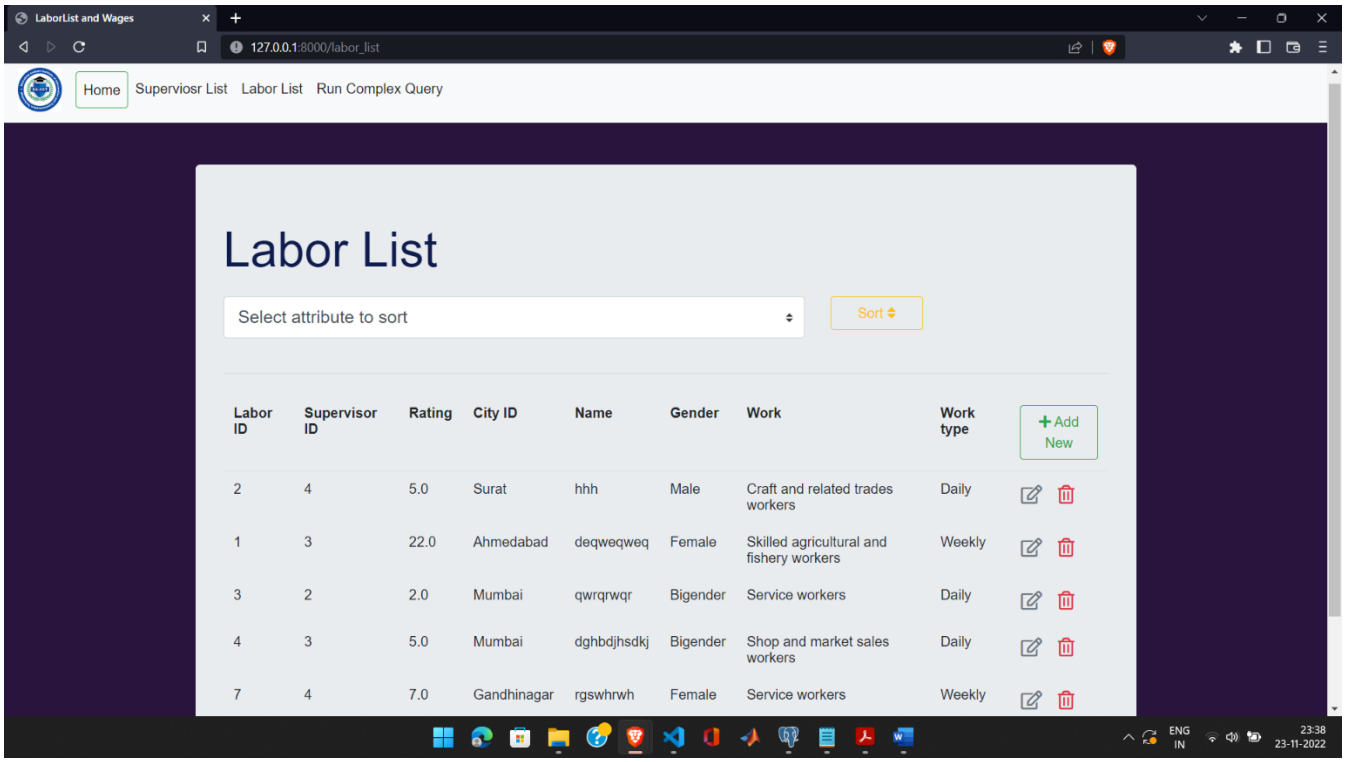
Sort function will give you facility for sorting data by any attribute :

For example :

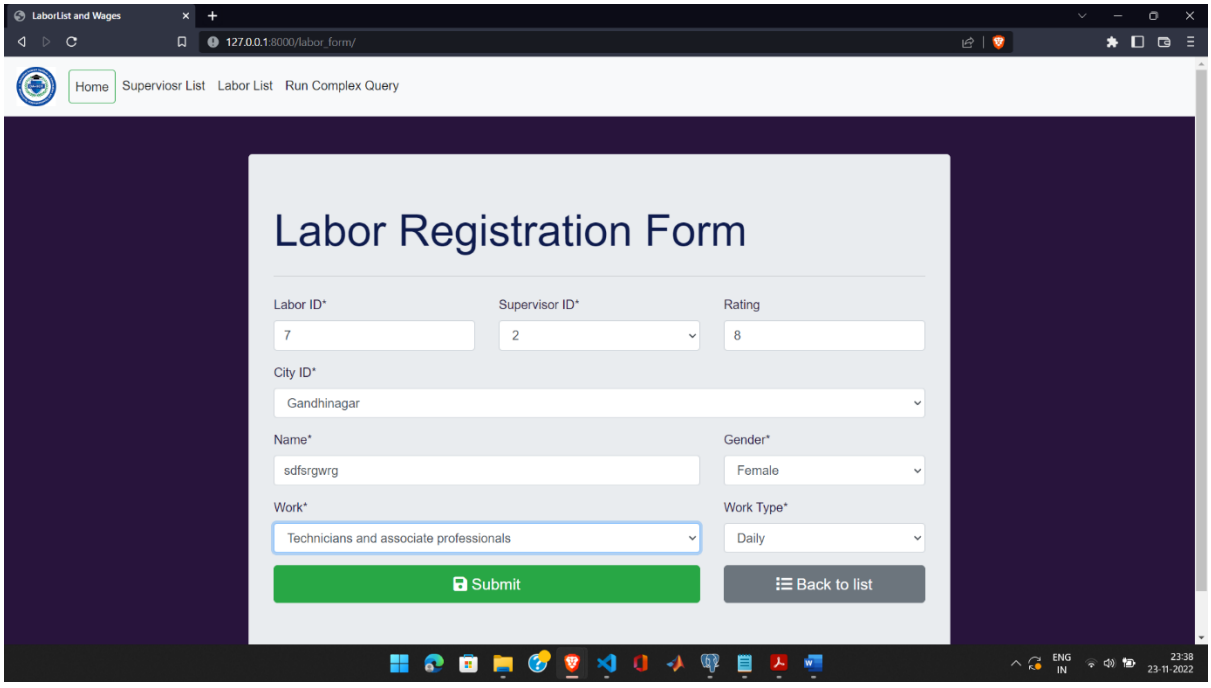


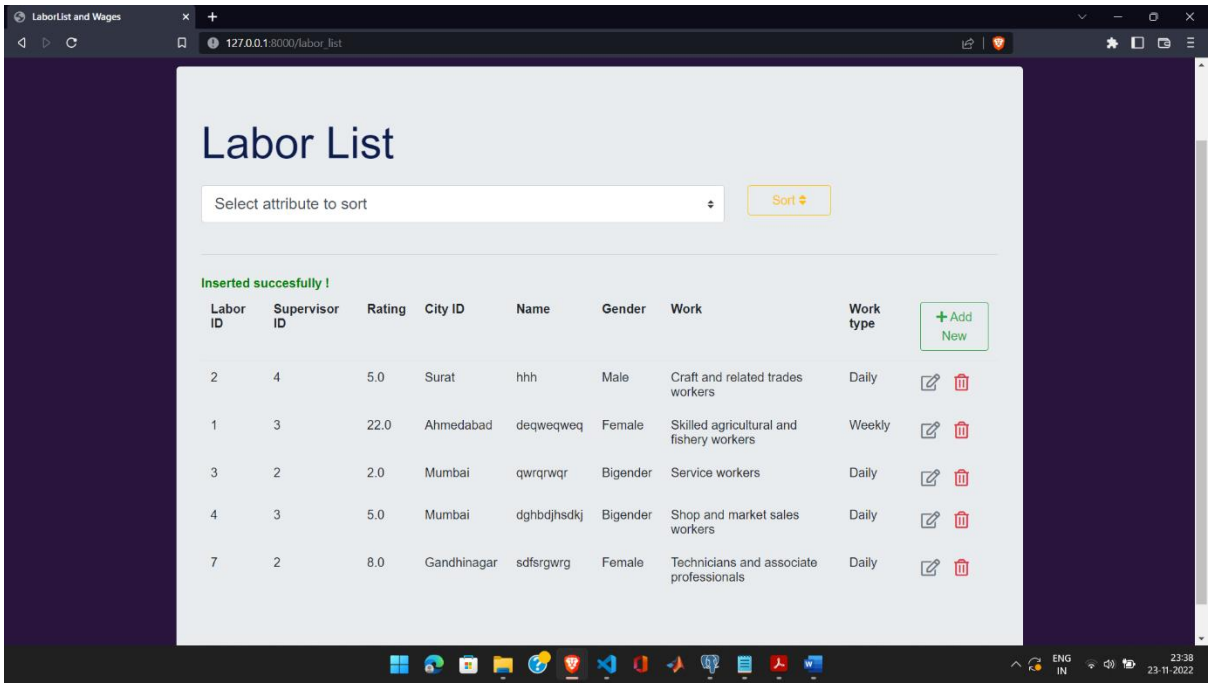


8. Supervisor Login/Laborer List :

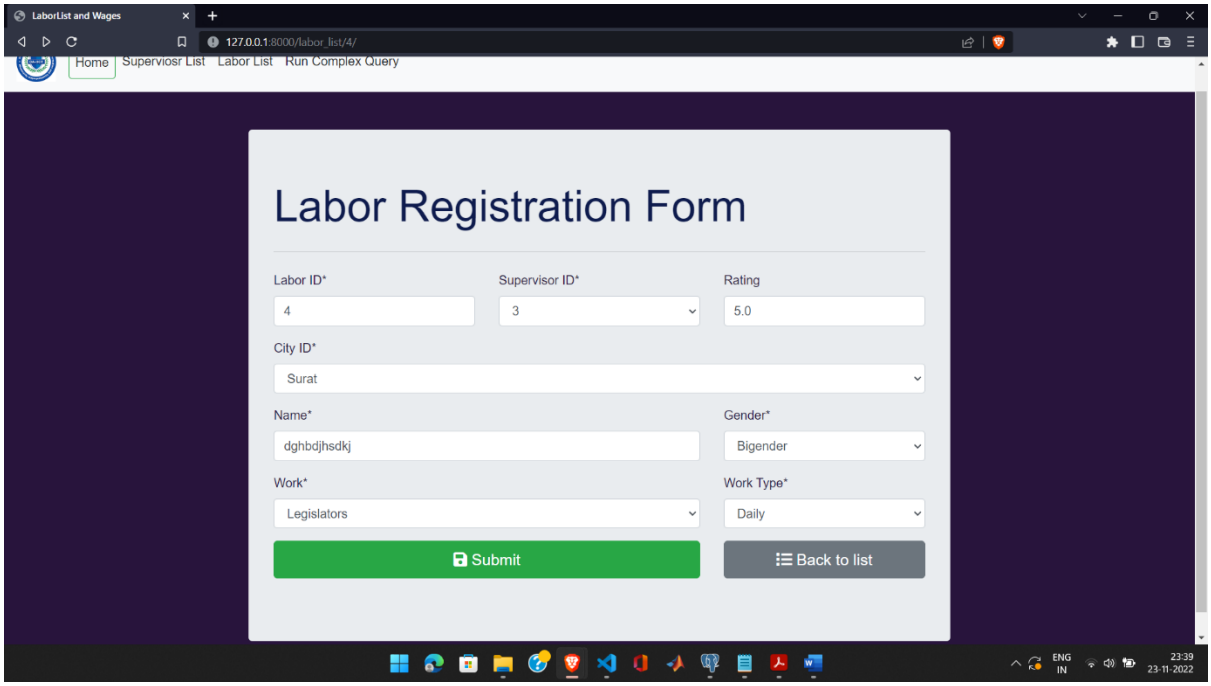


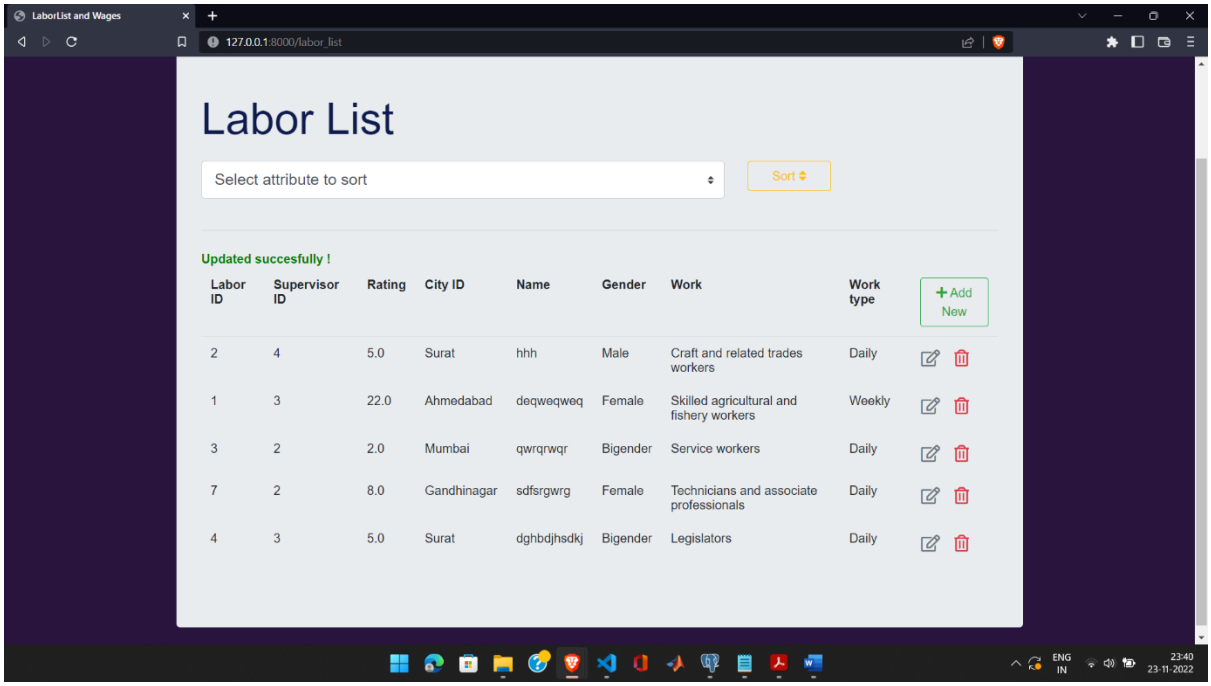
9. Insert Labor using Add new button:





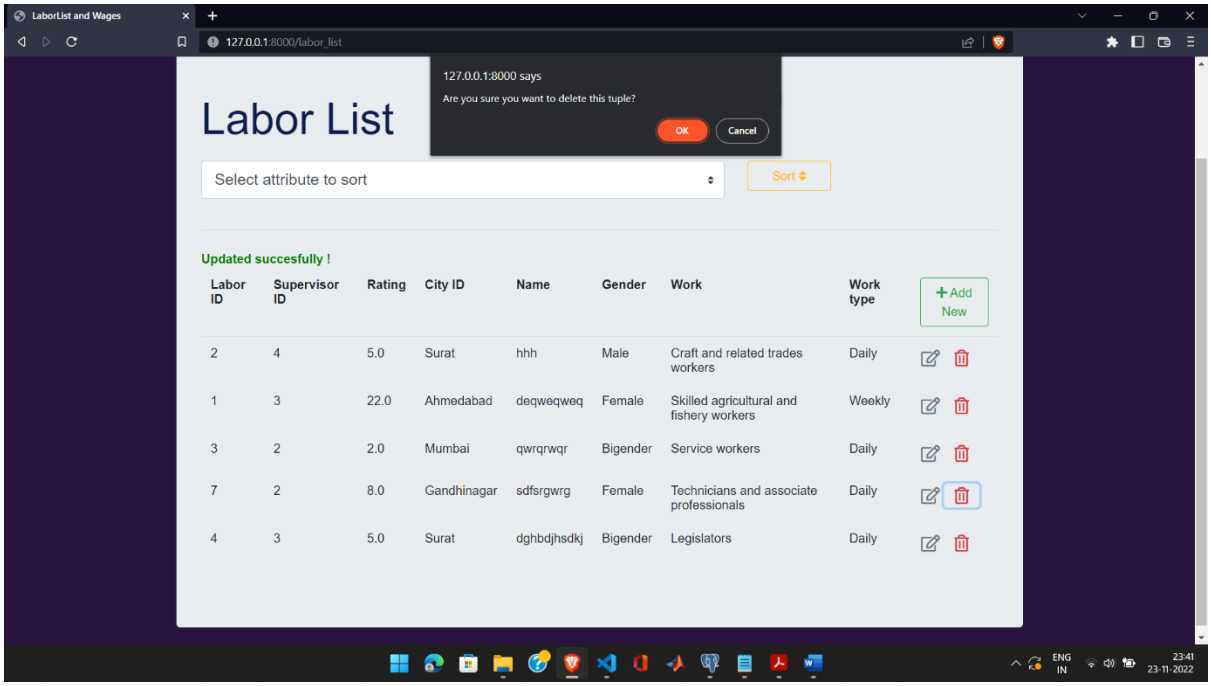
10. Update Labor details using the update button :

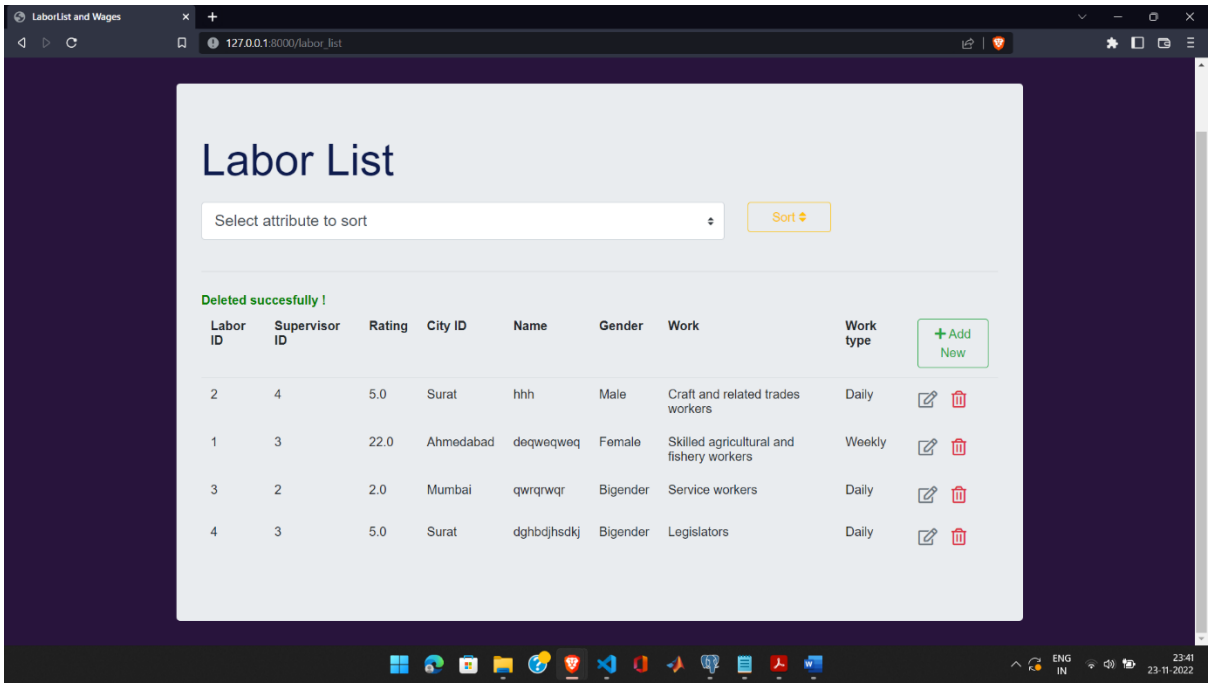




11. Delete Labor with trash button

(Delete will ask for conformation before deleting the data)

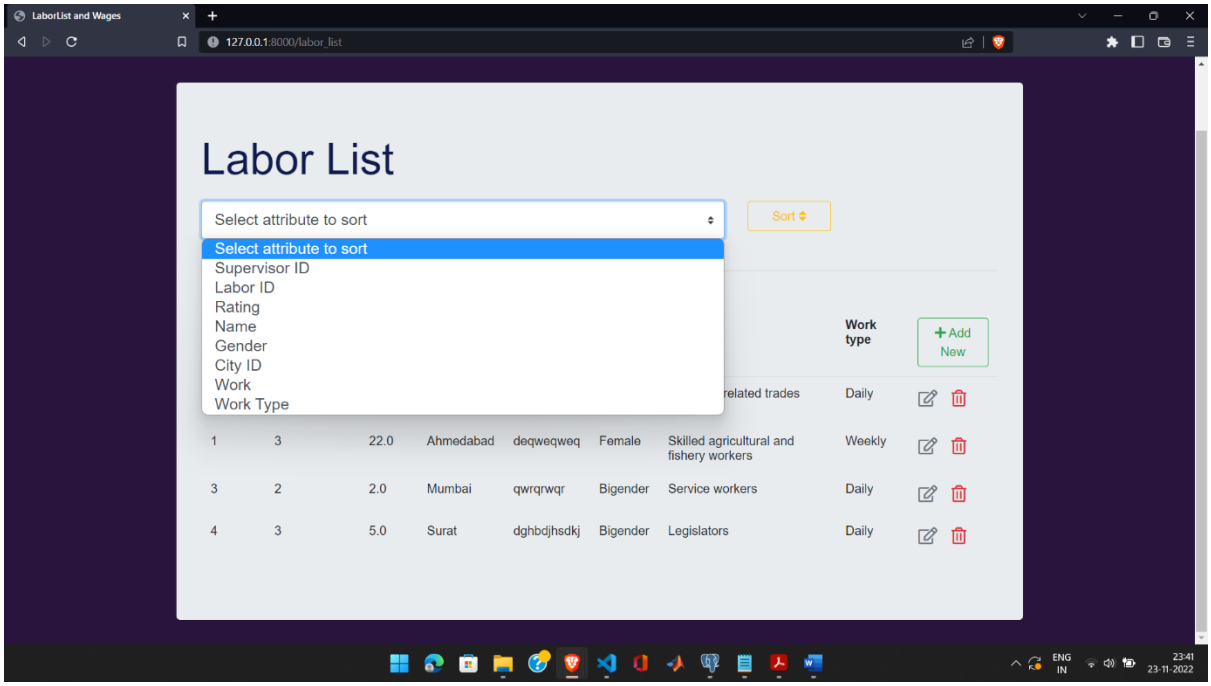


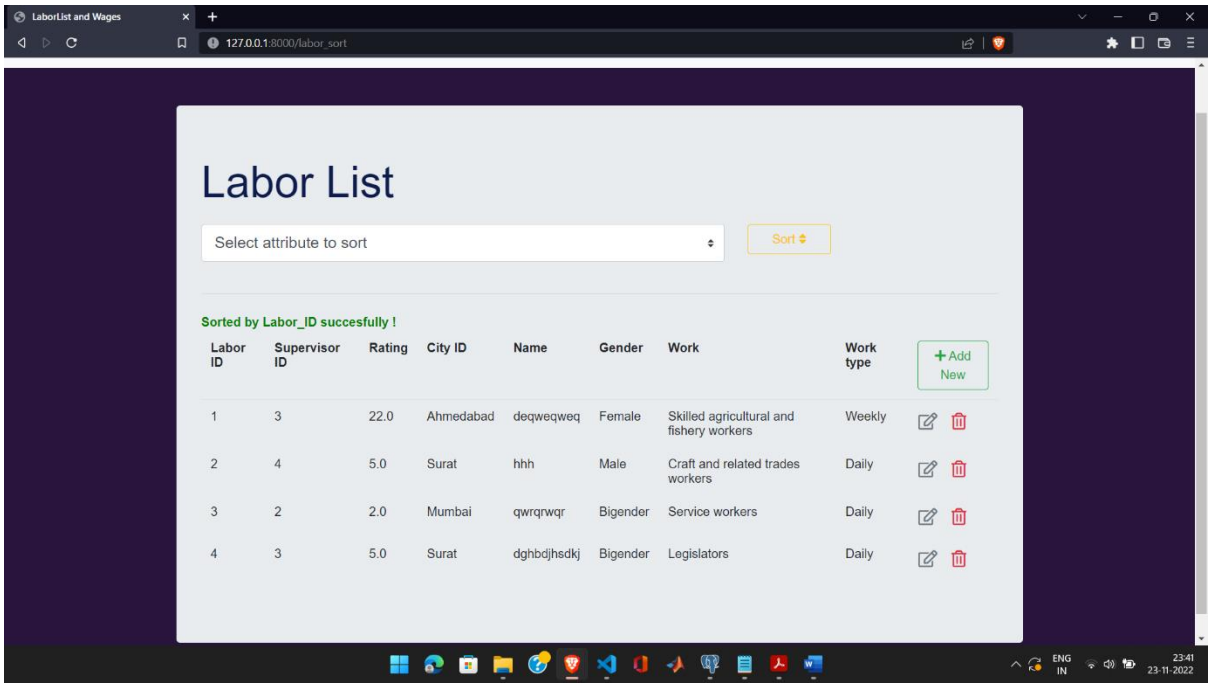


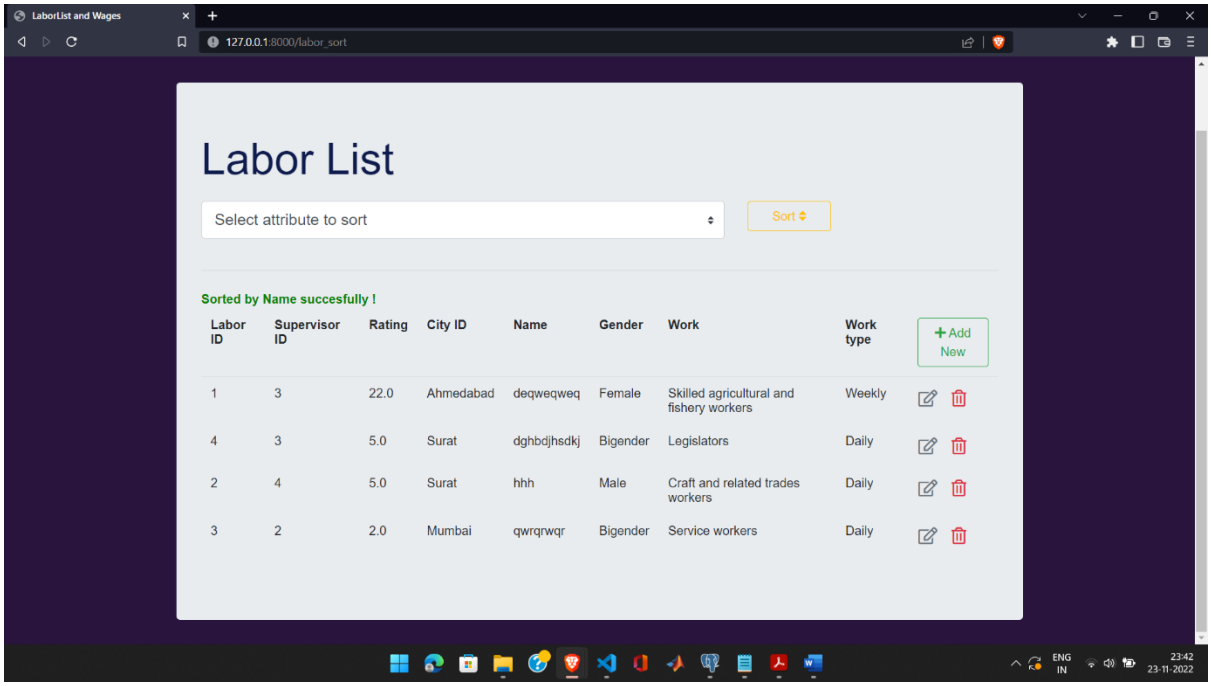
12.Sort function (Labor Data) :

Sort function will give you facility for sorting data by any attribute :

For example :


LaborList and Wages
127.0.0.1:8000/labor_list
Labor List
Select attribute to sort
Sort
Select attribute to sort
Supervisor ID
Labor ID
Rating
Name
Gender
City ID
Work
Work Type
Work type
+ Add New
related trades
Daily
1 3 22.0 Ahmedabad deqweqweq Female Skilled agricultural and fishery workers Weekly
3 2 2.0 Mumbai qwrqrwqr Bigender Service workers Daily
4 3 5.0 Surat dghbdjhsdkj Bigender Legislators Daily
ENG IN
23:41
23-11-2022

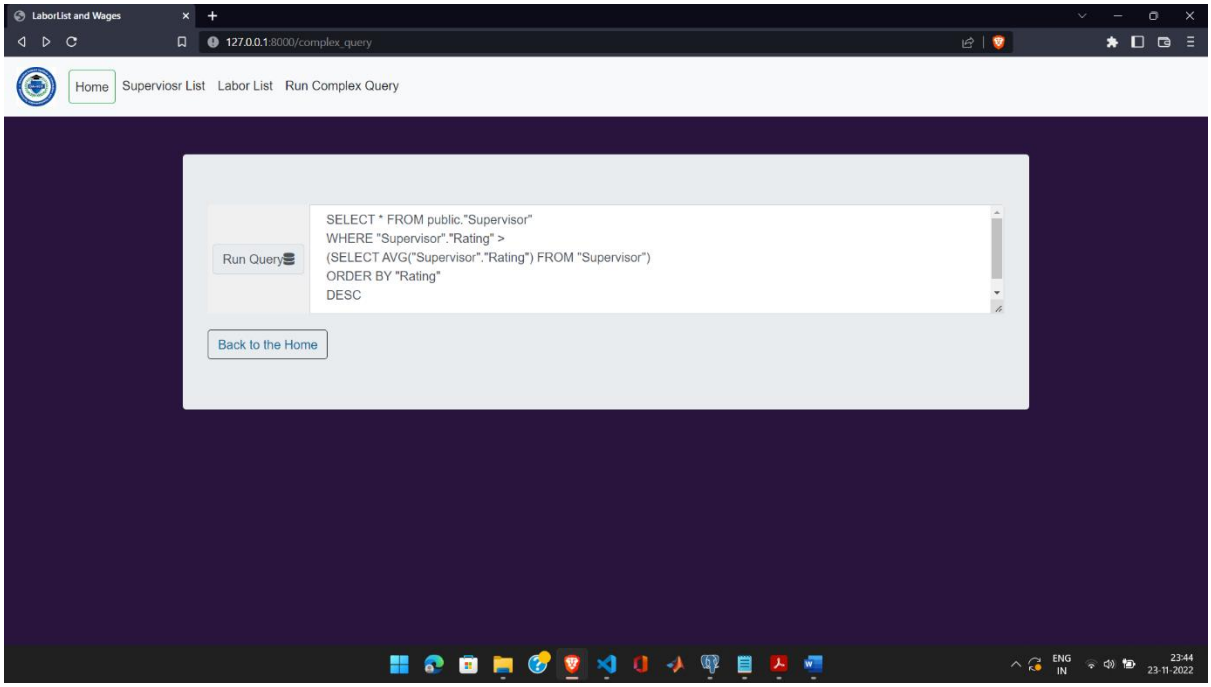

LaborList and Wages
127.0.0.1:8000/labor_sort
Labor List
Select attribute to sort
Sort
Sorted by Labor_ID succesfully !
Labor ID
Supervisor ID
Rating
City ID
Name
Gender
Work
Work type
+ Add New
1 3 22.0 Ahmedabad deqweqweq Female Skilled agricultural and fishery workers Weekly
2 4 5.0 Surat hhh Male Craft and related trades workers Daily
3 2 2.0 Mumbai qwrqrwqr Bigender Service workers Daily
4 3 5.0 Surat dghbdjhsdkj Bigender Legislators Daily
ENG IN
23:41
23-11-2022

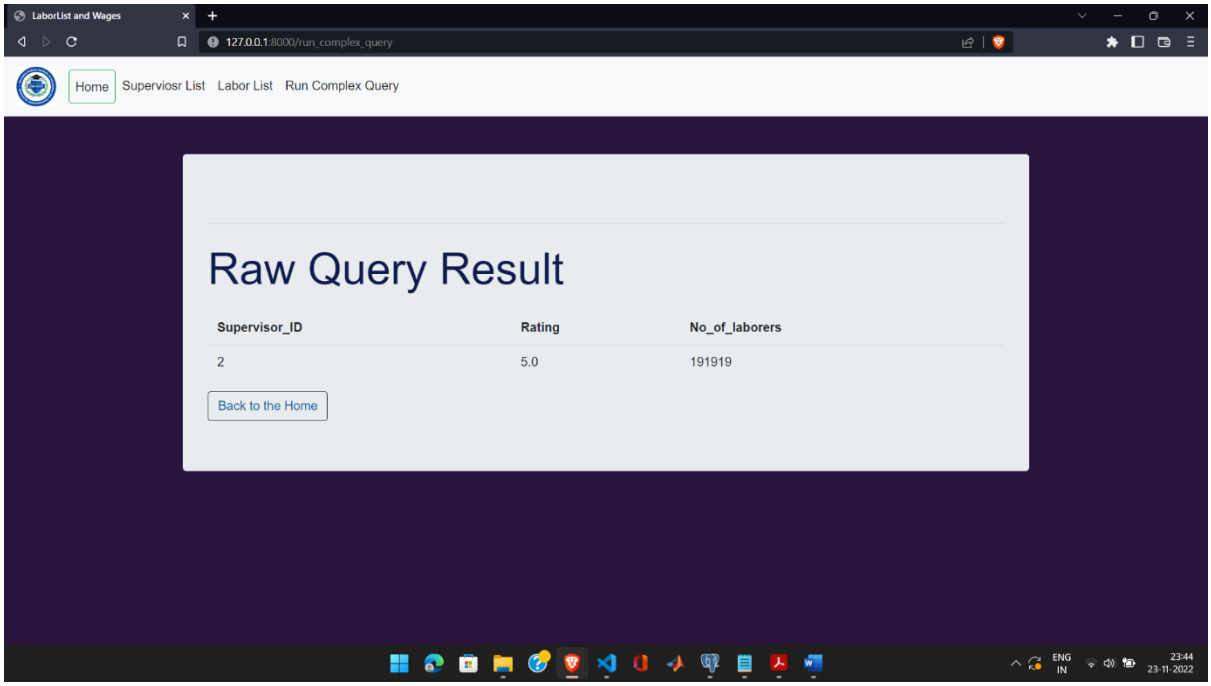


13. Run Complex Query :

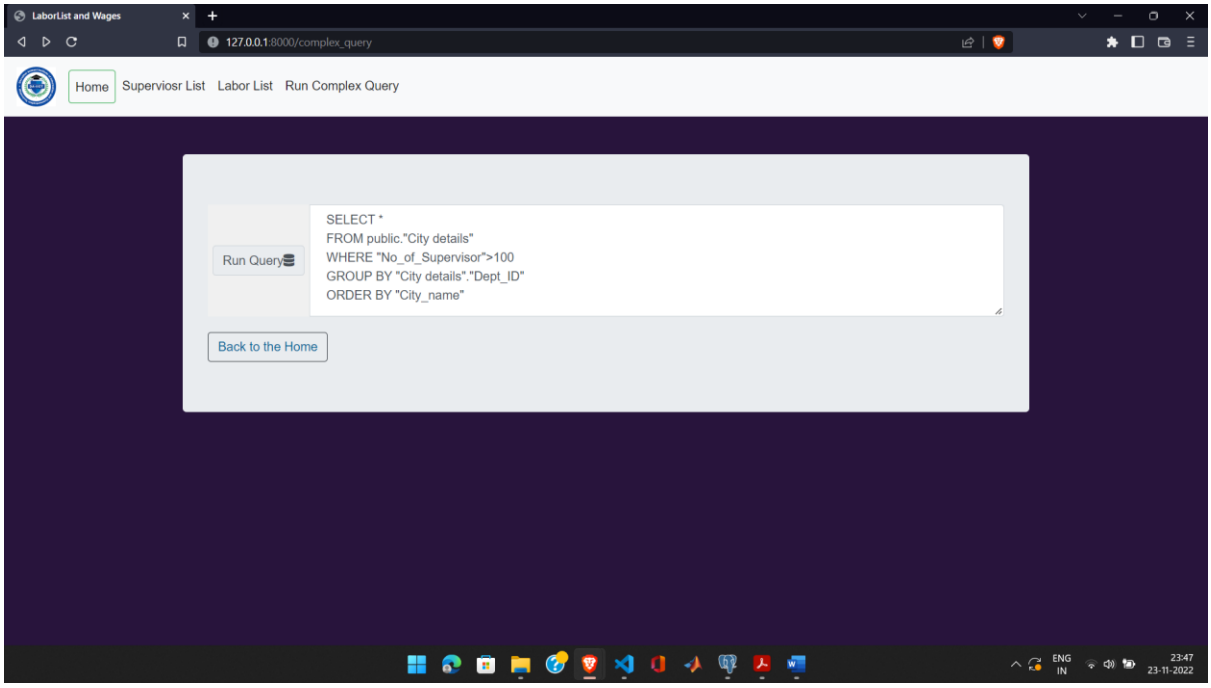
You can run any Query that you run in the database from this page :

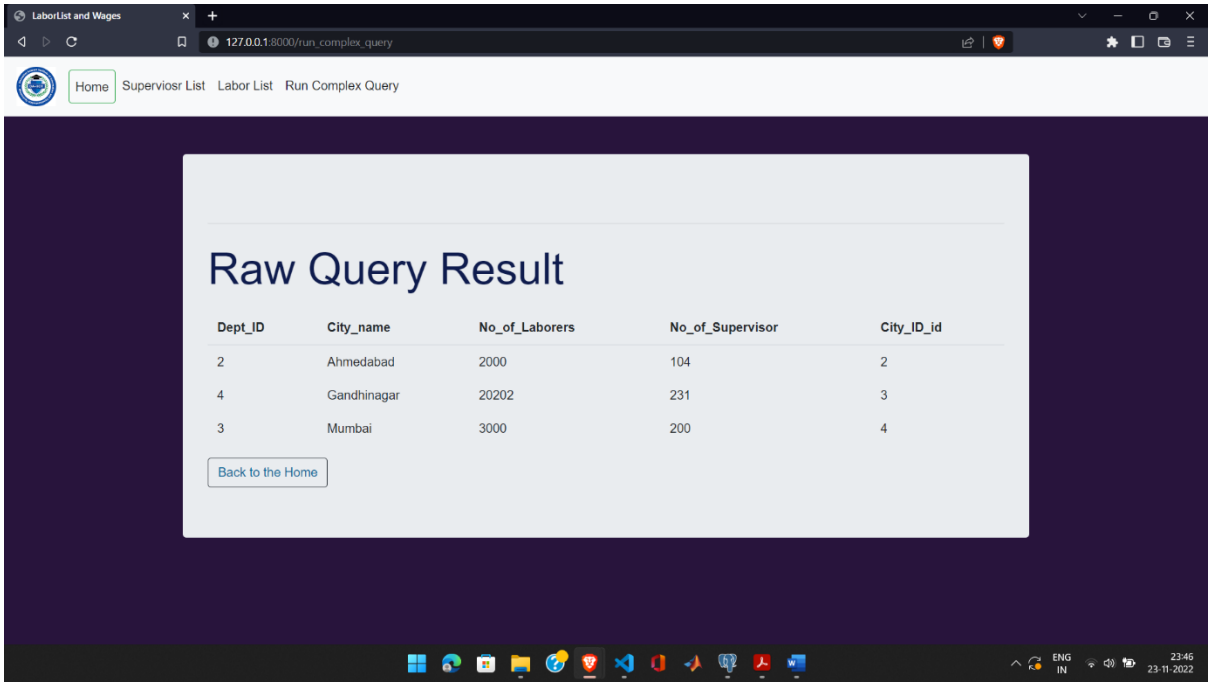
Example:




LaborList and Wages
127.0.0.1:8000/run_complex_query
Home
Superviosr List
Labor List
Run Complex Query
Raw Query Result
Supervisor_ID
Rating
No_of_laborers
2
5.0
191919
Back to the Home

Example 2 :


LaborList and Wages
127.0.0.1:8000/complex_query
Home
Superviosr List
Labor List
Run Complex Query
Run Query
SELECT *
FROM public."City details"
WHERE "No_of_Supervisor">100
GROUP BY "City details"."Dept_ID"
ORDER BY "City_name"
Back to the Home



GitHub Repository Link:

https://github.com/Devdeep-J-S/LaborList-and-Wages