Database Design and Applications (S2-22_SSABZG518)

Project
Payroll Management System

Date: Jul 22, 2023

Submitted To: Prof. Akanksha Bharadwaj {akanksha.bharadwaj@pilani.bits-pilani.ac.in}

Submitted By:

Name	Roll Number
Gunnidh kaur	2022AB12539
Kumari Dolly Singh	2022AB12522
Rashmi Khandelwal	2022AB12543
Suma Bhat	2022AB12529

Acknowledgement:

We are grateful to our Prof. Akanksha Bharadwaj, whose guidance, inspiration and constructive suggestions throughout the project has resulted in a successful completion of this project. Without their willing disposition, cooperation this project could not have been completed in due time.

We are also thankful to our Family, friends and colleagues for their cooperation and support in their own way.

Content

Abstract

The "Payroll Management System" is designed to automate the existing manual system with the help of computerized equipment and cutting-edge computer software, thereby meeting their needs and allowing their valuable data and information to be stored for a longer period of time with easy access and manipulation. The required software is easily accessible and straightforward to use. This online program can save and display electronic records while avoiding the creation of duplicate entries. The project addresses how to manage user data for improved speed and client services.

Introduction

The proposed project "Payroll Management System" was developed to overcome problems experienced when using manual processes. This program is intended to eliminate and, in some cases, reduce the challenges encountered by the current system. Furthermore, this system is customized to the company's specific needs in order to ensure smooth and efficient operations.

This project has been reduced as much as possible in order to minimize data entering issues. When invalid data is entered, it also displays an error notice. It is easy to use because no formal knowledge is required.
Every organization faces human resource challenges that must be overcome. Every company has different personnel and payroll administration needs. As a result, this is a unique Payroll Management System tailored to the organization's managerial requirements.

Purpose

The intent of this document is to explain the project's functionality and norms for Managing Employees and their Payroll. This document's intended audiences are developers and administrators. With the help of this system, the administrator now has all of the information at his fingertips and can exclusively make a logical chronology grounded on their requirements. Eventually, we can state that this system won't only automate the procedure but will also save the manager or administrator significant time that can be better spent. This will be an added advantage and power management based on their free time in addition to their usual activities.

Users of the Application

1. **Admin**

The Admin logs in with a valid username and password. Admin has the ability to add new employees, divisions, and pay grades to existing employees. It can also add new administrators and remove existing ones. The Admin can generate an employee's monthly paycheck automatically. Any previously recorded employee's earlier records are accessible to the administrator.

2. **Employee**

The Employee can login into their account which has been created by the administrator and using a valid username and password Employee and Admin module is Separated by the logical view of the database (how the data is perceived by end users) and the physical view (how the data is actually organized on storage media). Employees will be able to see all the deductions in their salary with respect to different components.

Software Description

Languages used

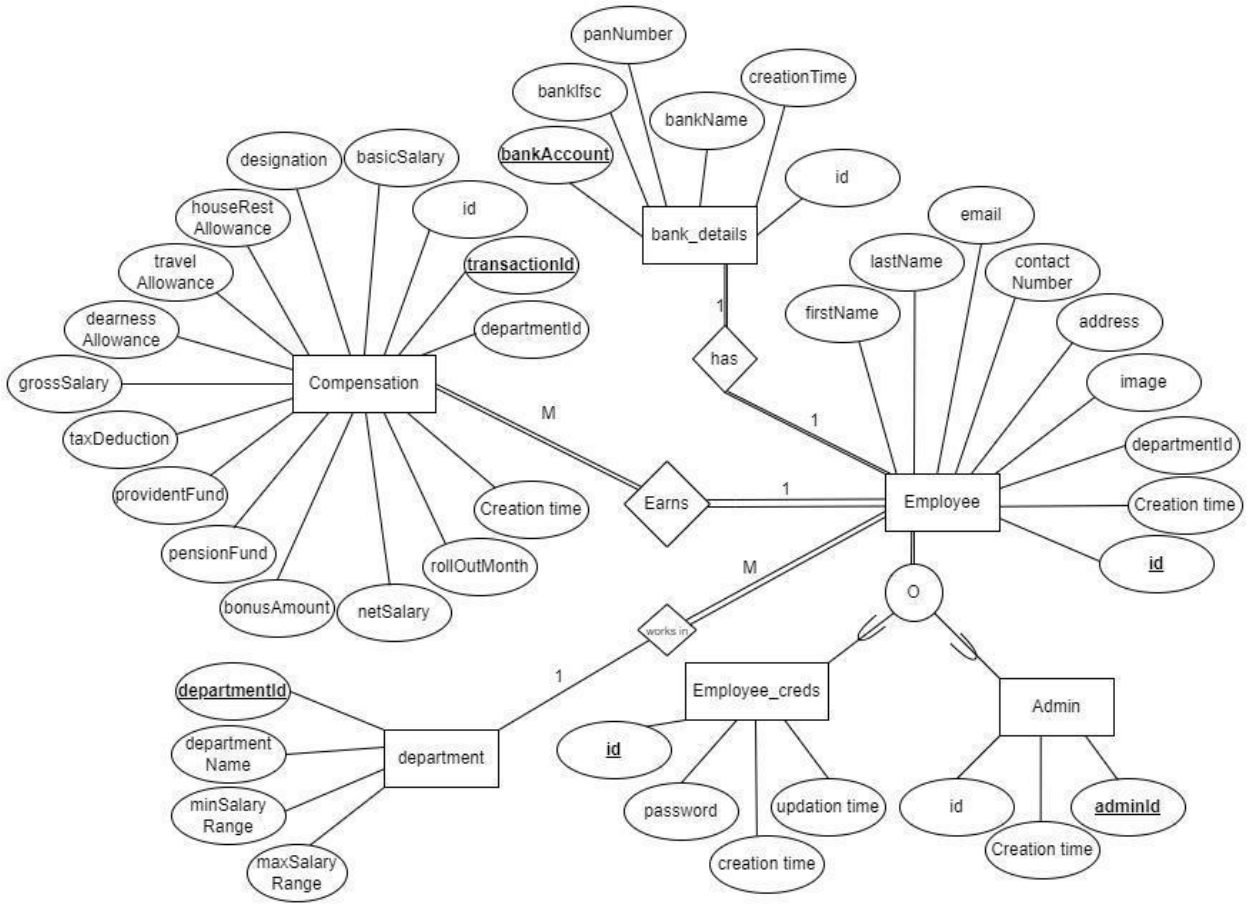
- **Languages:** React,Node JS
- **DBMS:** MySQL

Tools Used:

- **Editor Used:**
  - Visual studio code
  - XAMPP (phpmyadmin)server - It is a third-party tool to manage the tables and data inside the database. phpMyAdmin supports various types of operations on MariaDB and MySQL. The main purpose of phpMyAdmin is to handle the administration of MySQL over the web.
  - Draw.io

- **Operating System:** Windows 10

EER Diagram


panNumber
bankIfsc
creationTime
bankName
bankAccount
id
bank_details
designation
basicSalary
houseRest Allowance
id
travel Allowance
transactionId
dearness Allowance
departmentId
grossSalary
Compensation
taxDeduction
providentFund
Creation time
pensionFund
rollOutMonth
bonusAmount
netSalary
1
has
1
M
Earns
1
Employee
email
lastName
contact Number
firstName
address
image
departmentId
Creation time
id
M
O
works in
1
departmentId
department Name
minSalary Range
maxSalary Range
department
Employee_creds
id
password
updation time
creation time
Admin
id
adminId
Creation time

EER Description

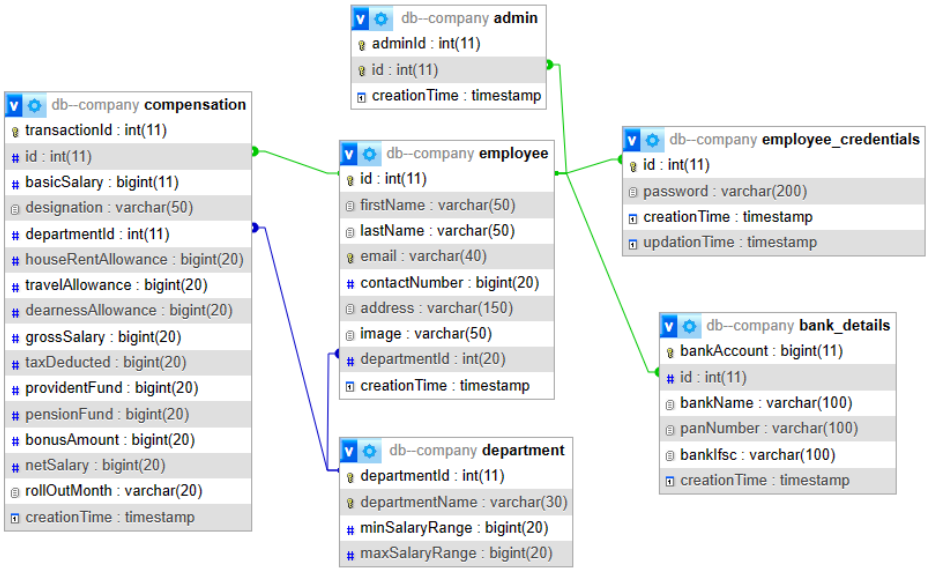
- **Generalization:** Admin and Employee_credential tables are subclasses of the Employee table as they are inheriting its attributes and Relationship.This is a overlapping and total specialization as justified below:
- **Overlapping:** An employee (in superclass) can have his/her data in both subclasses Employee and Admin
- **Total:** Every Entity in the Employee table must be having their data in either or both subclasses.
- **Cardinality:** For the relation employee earns compensation, We have 1 to many cardinality. Every employee will have multiple salary credits (on a monthly basis) but one salary credit is related to only one employee. Also this relation is a case of total participation. Each employee will have a record in compensation and also every record in compensation will be associated with an employee.

For the relation employee works in the Department , we have many to 1 relation as one employee can work for one department but one department will have multiple employees.

For the relation employee has bank_details, we have 1 to 1 relation as each employee will have 1 record in the bank_details table and one bank data is related to one employee only.

- **Keys:**
- **Employee:** id is the primary key and is an auto increment attribute
- **Department** : DepartmentID is the primary key and id is the Foreign key referenced as Foreign key
- **Bank_details** : bankAccount is the primary key and id from employee table is referenced as Foreign key
- **Compensation** : transactionId is the primary key and id from employee table is referenced as Foreign key
- **Employee_creds** : Since there is one to one relationship between employee and this table , the foreign key id from employee is used as Primary key
- **Admin** : AdminId is the primary key id from the employee table and is referenced as Foreign key.

Schema Diagram



Mapping Rules

Employee Table

- Entity: employee
- Table Name: employee
- Attributes: id, firstName, lastName, email, contactNumber, address, image, bankAccount, departmentId, creationTime

id	firstName	lastName	email	contactNumber	address	image	bankAccount	departmentId	creationTime

Admin Table

- Entity: admin
- Table Name: admin
- Attributes: adminId, id, creationTime

adminId	id	creationTime

Department Table

- Entity: department
- Table Name: department
- Attributes: departmentId, departmentName, minSalaryRange, maxSalaryRange

<u>departmentId</u>	departmentName	minSalaryRange	maxSalaryRange

Bank Detail

- Entity: bank_details
- Table Name: bank_details
- Attributes: bankAccount, id, bankName, panNumber, bankIfsc, creationTime

<u>bankAccount</u>	id	bankName	panNumber	bankIfsc	creationTime

Compensation Table

- Entity: compensation
- Table Name: compensation
- Attributes: transactionId, id, basicSalary, designation, departmentId, houseRentAllowance, travelAllowance, dearnessAllowance, grossSalary, taxReduction, providentFund, pensionFund, bonusAmount, netSalary, rollOutMonth, creationTime

<u>transactionId</u>	id	basicSalary	designation	departmentId	houseRentAllowance

travelAllowance	dearnessAllowance	grossSalary	taxReduction	providentFund

pensionFund	bonusAmount	netSalary	rollOutMonth	creationTime

Snippets of codes and working model

Structure of the **database**

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
bank_details	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 KiB	-
compensation	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	48.0 KiB	-
department	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
employee	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	80.0 KiB	-
employee_credentials	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-

Structure of the table **admin**

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	adminId	int(11)			No	*None*			Change Drop More
2	id	int(11)			No	*None*			Change Drop More
3	creationTime	timestamp			No	current_timestamp()			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after creationTime Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	adminId	0	A	No	
Edit Rename Drop	id	BTREE	Yes	No	id	0	A	No	

Structure of the table **employee_credentials**

Server: 127.0.0.1 » Database: db--company » Table: employee_credentials

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	password	varchar(200)	utf8mb4_general_ci		No	$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPIl7wZ6JpgENTQKm			Change Drop More
3	creationTime	timestamp			No	current_timestamp()			Change Drop More
4	updationTime	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after updationTime Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	8	A	No	

Structure of the table **compensation**

Server: 127.0.0.1 » Database: db–company » Table: compensation

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	transactionId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	id	int(11)			No	None			Change Drop More
3	basicSalary	bigint(11)			No	None			Change Drop More
4	designation	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	departmentId	int(11)			No	None			Change Drop More
6	houseRentAllowance	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
7	travelAllowance	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
8	dearnessAllowance	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
9	grossSalary	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
10	taxDeducted	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
11	providentFund	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
12	pensionFund	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
13	bonusAmount	bigint(20)			Yes	NULL			Change Drop More
14	netSalary	bigint(20)			Yes	NULL		STORED GENERATED	Change Drop More
15	rollOutMonth	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
16	creationTime	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after creationTime Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	transactionId	10	A	No	
Edit Rename Drop	compensation_ibfk_1	BTREE	No	No	id	10	A	No	
Edit Rename Drop	departmentId	BTREE	No	No	departmentId	10	A	No	

Structure of the table **department**

Server: 127.0.0.1 » Database: db–company » Table: department

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	departmentId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	departmentName	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	minSalaryRange	bigint(20)			No	None			Change Drop More
4	maxSalaryRange	bigint(20)			No	None			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after maxSalaryRange Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	departmentId	4	A	No	
Edit Rename Drop	department_name	BTREE	Yes	No	departmentName	4	A	No	

Structure of the table **employee**

Server: 127.0.0.1 » Database: db--company » Table: employee

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	firstName	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	lastName	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	email	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More
5	contactNumber	bigint(20)			No	None			Change Drop More
6	address	varchar(150)	utf8mb4_general_ci		No	None			Change Drop More
7	image	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
8	departmentId	int(20)			No	None			Change Drop More
9	creationTime	timestamp			No	current_timestamp()			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after creationTime Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	id	10	A	No	
Edit Rename Drop	email	BTREE	Yes	No	email	10	A	No	
Edit Rename Drop	department_id	BTREE	No	No	departmentId	10	A	No	

Structure of the table **bank_details**

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	bankAccount	bigint(11)			No	None			Change Drop More
2	id	int(11)			No	None			Change Drop More
3	bankName	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	panNumber	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	bankIfsc	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	creationTime	timestamp			No	current_timestamp()			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns Remove from central columns

Print Propose table structure Track table Move columns Normalize

Add 1 column(s) after creationTime Go

Indexes

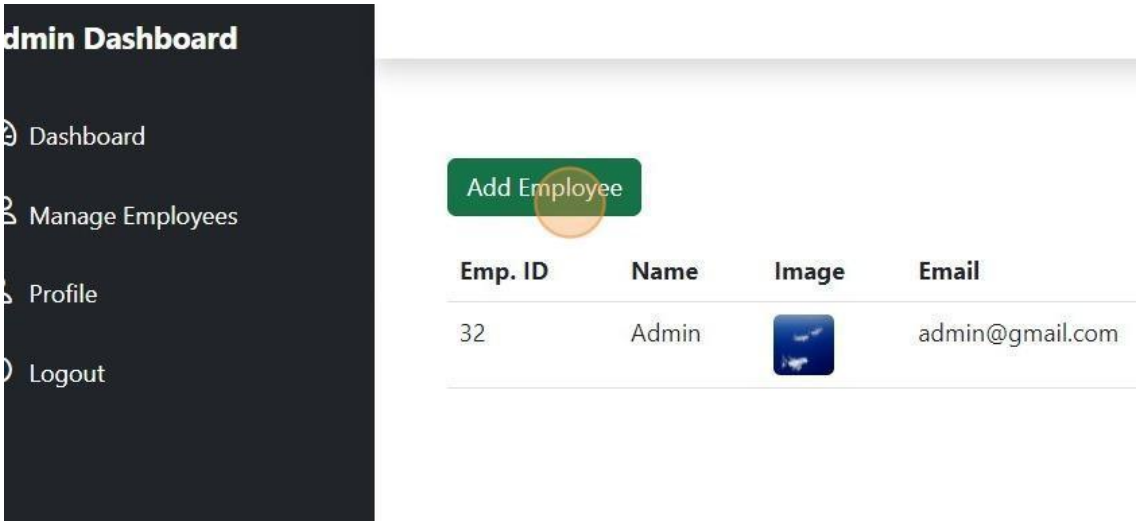
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	bankAccount	10	A	No	
Edit Rename Drop	id	BTREE	No	No	id	10	A	No	

Working Model FrontEnd

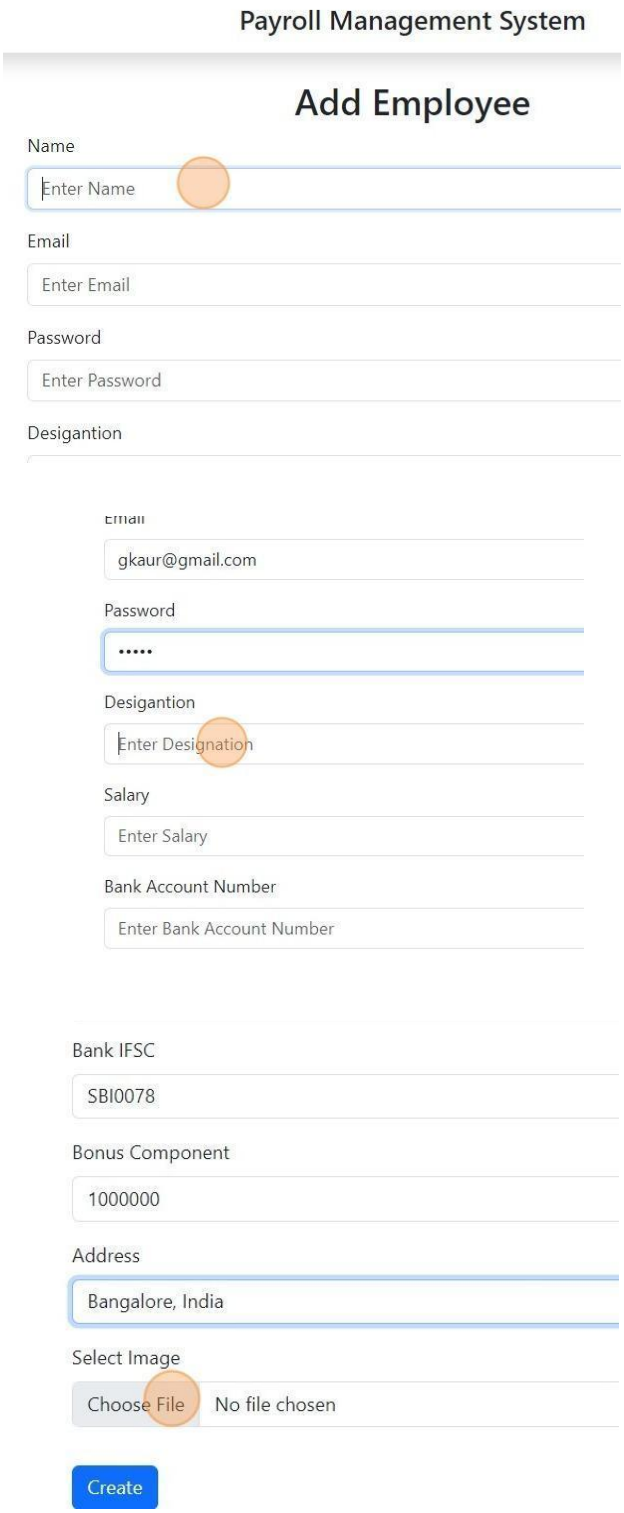
- Select the type of login

- Enter the credentials

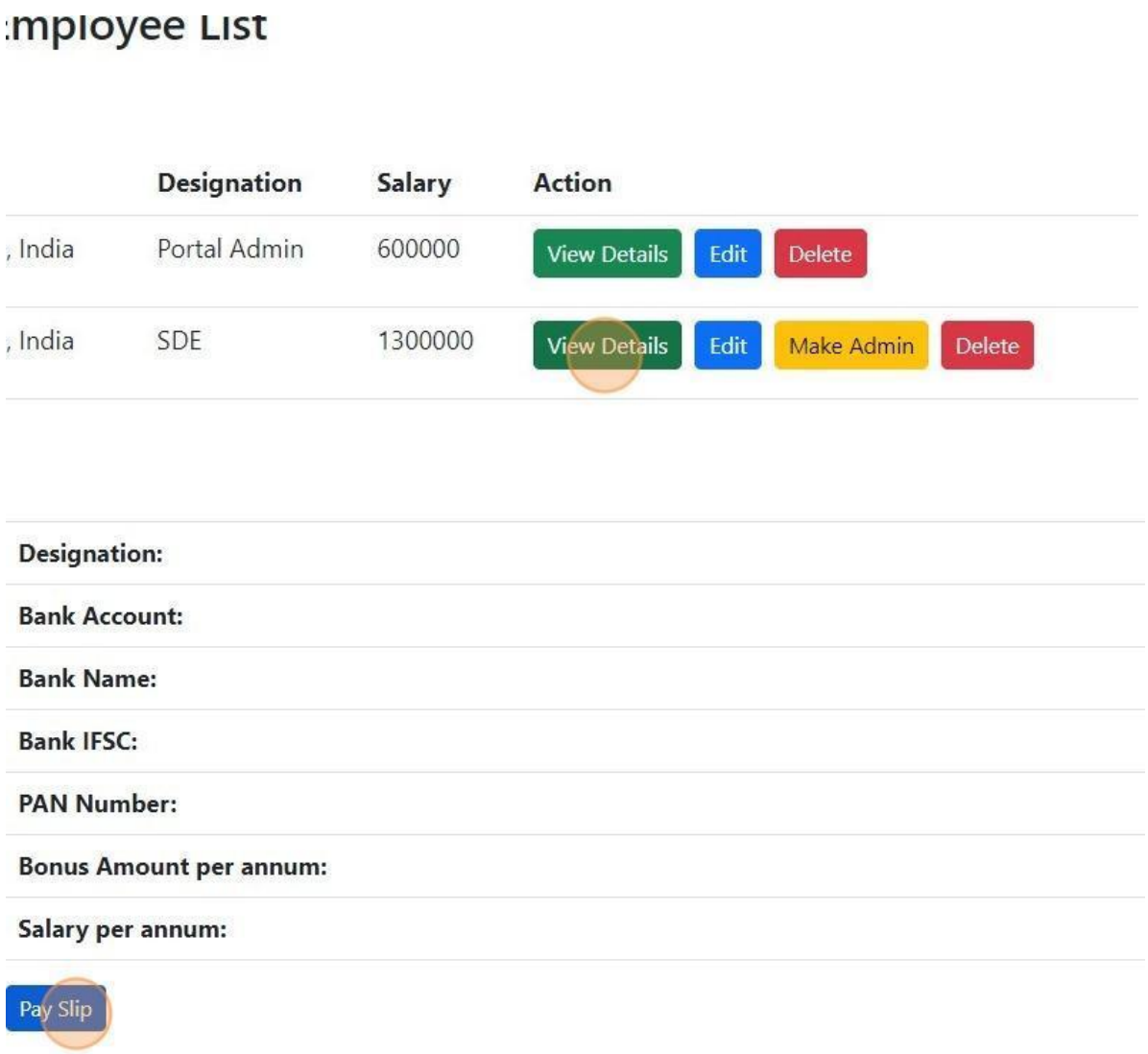
- As an admin you can add employees



- Fill out the basic information required to add an employee and click on create



- Employee details and payslips can be viewed from here:



Employee can login into their accounts and can see their details as well as their monthly payslip

Salary-slip created on July 22, 2023 at 6:48:40 AM GMT+5:30

Employee ID:	41
Name:	Dolly
Email:	singh@gmail.om
Address:	Bangalore, India
Salary:	₹ 141666.67
Designation:	SDE-2
HRA:	₹ 28333.33
DA:	₹ 25500.00
TA:	₹ 21250.00
PF:	₹ 17000.00
Pension Fund:	₹ 11333.33
Gross Salary:	₹ 191250.00
Net Salary:	₹ 115104.17
Salary credited to Bank Account:	7657659
PAN associated with deduction:	777GTRF0CCi

DDL Queries:

Query used for creation of different tables

admin

```
CREATE TABLE `admin` (
  `adminId` int(11) NOT NULL,
  `id` int(11) NOT NULL,
  `creationTime` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

bank_details

```
--
-- Table structure for table `bank_details`
--

CREATE TABLE `bank_details` (
  `bankAccount` bigint(11) NOT NULL,
  `id` int(11) NOT NULL,
  `bankName` varchar(100) NOT NULL,
  `panNumber` varchar(100) NOT NULL,
  `bankIfsc` varchar(100) NOT NULL,
  `creationTime` timestamp NOT NULL DEFAULT current_timestamp(),
  FOREIGN KEY (id) REFERENCES employee (id),
  PRIMARY KEY (bankAccount)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
```

department

```
-- Table structure for table `department`
--

CREATE TABLE `department` (
  `departmentId` int(11) NOT NULL,
  `departmentName` varchar(30) NOT NULL,
  `minSalaryRange` bigint(20) NOT NULL,
  `maxSalaryRange` bigint(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
```

compensation

```
-- Table structure for table `compensation`
--

CREATE TABLE `compensation` (
  `transactionId` int(11) NOT NULL,
  `id` int(11) NOT NULL,
  `basicSalary` bigint(11) NOT NULL,
  `designation` varchar(50) NOT NULL,
  `departmentId` int(11) NOT NULL,
  `houseRentAllowance` bigint(20) GENERATED ALWAYS AS (`basicSalary` * 0.2) STORED,
  `travelAllowance` bigint(20) GENERATED ALWAYS AS (`basicSalary` * 0.15) STORED,
  `dearnessAllowance` bigint(20) GENERATED ALWAYS AS (`basicSalary` * 0.18) STORED,
  `grossSalary` bigint(20) GENERATED ALWAYS AS (`basicSalary` + `basicSalary` * 0.2 + `basicSalary` * 0.15) STORED,
  `taxDeducted` bigint(20) GENERATED ALWAYS AS ((`basicSalary` + `basicSalary` * 0.2 + `basicSalary` * 0.15) * 0.25) STORED,
  `providentFund` bigint(20) GENERATED ALWAYS AS (`basicSalary` * 0.12) STORED,
  `pensionFund` bigint(20) GENERATED ALWAYS AS (`basicSalary` * 0.08) STORED,
  `bonusAmount` bigint(20) DEFAULT NULL,
  `netSalary` bigint(20) GENERATED ALWAYS AS (`basicSalary` + `basicSalary` * 0.2 + `basicSalary` * 0.15 - (`basicSalary` + `basicSalary` * 0.2 + `basicSalary` * 0.15) * 0.25 - `basicSalary` * 0.12 - `basicSalary` * 0.08 + coalesce(`bonusAmount`,0)) STORED,
  `rollOutMonth` varchar(20) NOT NULL,
  `creationTime` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

employee

```
CREATE TABLE `employee` (
  `id` int(11) NOT NULL,
  `firstName` varchar(50) NOT NULL,
  `lastName` varchar(50) NOT NULL,
  `email` varchar(40) NOT NULL,
  `contactNumber` bigint(20) NOT NULL,
  `address` varchar(150) NOT NULL,
  `image` varchar(50) NOT NULL,
  `departmentId` int(20) NOT NULL,
  `creationTime` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

employee_credentials

```
--
-- Table structure for table `employee_credentials`
--

CREATE TABLE `employee_credentials` (
  `id` int(11) NOT NULL,
  `password` varchar(200) NOT NULL DEFAULT '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm',
  `creationTime` timestamp NOT NULL DEFAULT current_timestamp(),
  `updationTime` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Query used for foreign key cascading

Alter table queries generated for foreign and primary keys:

```sql
-- Constraints for dumped tables
--

--
-- Constraints for table `admin`
--
ALTER TABLE `admin`
  ADD CONSTRAINT `admin_ibfk_1` FOREIGN KEY (`id`) REFERENCES `employee` (`id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `bank_details`
--
ALTER TABLE `bank_details`
  ADD CONSTRAINT `bank_details_ibfk_1` FOREIGN KEY (`bankAccount`) REFERENCES `employee` (`bankAccount`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `compensation`
--
ALTER TABLE `compensation`
  ADD CONSTRAINT `compensation_ibfk_1` FOREIGN KEY (`id`) REFERENCES `employee` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `compensation_ibfk_2` FOREIGN KEY (`departmentId`) REFERENCES `department` (`departmentId`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `employee`
--
ALTER TABLE `employee`
  ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`departmentId`) REFERENCES `department` (`departmentId`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `employee_credentials`
--
ALTER TABLE `employee_credentials`
  ADD CONSTRAINT `employee_credentials_ibfk_1` FOREIGN KEY (`id`) REFERENCES `employee` (`id`) ON DELETE CASCADE;
COMMIT;
```

DML Queries

Query used to insert data into tables

```sql
-- Dumping data for table `bank_details`
--

INSERT INTO `bank_details` (`id`,`bankAccount`, `bankName`, `panNumber`, `bankIfsc`, `creationTime`) VALUES
(48,23456712, 'SBI', 'SBI67890', 'SBIN003', '2023-07-22 03:44:11'),
(46,334567443, 'ICICI Bank', 'ABCDE1234F', 'ICIC000001', '2023-07-22 03:44:11'),
(45,345464764, 'State Bank of India', 'FGHIJ5678K', 'SBIN000002', '2023-07-22 03:44:11'),
(49,456789123, 'Bank of America', 'BOA98765', 'BOA0011', '2023-07-22 03:44:11'),
(50,456789124, 'Chase Bank', 'CHASE54321', 'CHASE002', '2023-07-22 03:44:11'),
(41,7657659, 'HDFC', '777GTRF0CCI', 'HDFC0038', '2023-07-22 01:01:18'),
(44,543224556, 'Chase Bank', 'RSTUV3456W', 'CHAS000004', '2023-07-22 03:44:11'),
(43,765412221, 'HDFC', 'HQ234BN00H', 'HDFC0034', '2023-07-22 01:17:42'),
(32,2147483647, 'Axis Bank', 'HQ435GH89', 'AXIS009', '2023-07-21 16:13:10'),
(47,21474003646, 'ICICI', 'ICICI12345', 'ICIC0022', '2023-07-22 03:44:11');
```

```sql
INSERT INTO `admin` (`adminId`, `id`, `creationTime`) VALUES
(1, 32, '2023-07-21 16:13:22');
```

```sql
-- Dumping data for table `compensation`
--

INSERT INTO `compensation` (`transactionId`, `id`, `basicSalary`, `designation`, `departmentId`, `bonusAmount`, `rollOutMonth`, `creationTime`) VALUES
(1, 32, 1000000, 'Admin', 1, 50000, 'January', '2023-07-22 10:26:17'),
(9, 41, 2300000, 'Senior CA', 2, NULL, 'August', '2023-07-22 10:26:37'),
(10, 43, 1300000, 'SDE2', 3, 300000, '', '2023-07-22 10:26:55'),
(12, 44, 1100000, 'Admin', 2, 50000, 'July', '2023-07-22 10:32:38'),
(13, 45, 1200000, 'Head of Sales', 1, 7000, 'July', '2023-07-22 10:29:44'),
(14, 46, 1300000, 'HR Intern', 4, 50000, 'July', '2023-07-22 10:32:56'),
(15, 47, 1400000, 'System Engineer', 3, 5000, 'July', '2023-07-22 10:33:02'),
(16, 48, 1500000, 'Resource Specialist', 4, 50000, 'July', '2023-07-22 10:33:05'),
(17, 49, 1600000, 'HR Analyst', 4, 40000, 'July', '2023-07-22 10:33:07'),
(18, 50, 1700000, 'Procurement Specialist', 1, 45000, 'July', '2023-07-22 10:33:11');
```

```sql
-- Dumping data for table `department`
--

INSERT INTO `department` (`departmentId`, `departmentName`, `minSalaryRange`, `maxSalaryRange`) VALUES
(1, 'Sales', 350000, 2500000),
(2, 'Accounts', 800000, 15000000),
(3, 'Engineering', 1000000, 35000000),
(4, 'HR', 900000, 6000000);
```

```sql
INSERT INTO `employee` (`id`, `firstName`, `lastName`, `email`, `contactNumber`, `address`, `image`, `departmentId`, `creationTime`) VALUES
(32, 'Admin', 'Root', 'admin@gmail.com', 0, 'Bangalore, India', 'image_1689955990207.jpeg', 1, '2023-07-21 16:13:10'),
(41, 'Suma', 'Bhat', 'suma@gmail.com', 9999035299, 'Bangalore, India', 'image_1689987678635.jpg', 2, '2023-07-22 01:01:18'),
(43, 'Gunnidh', 'Kaur', 'gkaur@gmail.com', 7901808055, 'Bangalore, India', 'image_1689988662059.jpg', 3, '2023-07-22 01:17:42'),
(44, 'Joe', 'Biden', 'joeone@gamil.com', 1234567, 'India', '', 2, '2023-07-22 09:03:01'),
(45, 'Joe', 'Nicholas', 'joetwo@gmail.com', 6738958993, 'US', '', 1, '2023-07-22 09:03:01'),
(46, 'Priya', 'Ray', 'newuser@gmail.com', 356678544, 'UK', '12356774', 4, '2023-07-22 09:04:32'),
(47, 'Dolly', 'Singh', 'dolly@gmail.com', 345778445, 'Bangalore', '4456743', 3, '2023-07-22 09:14:11'),
(48, 'Rashmi', 'Khandelwal', 'rashmi@gmail.com', 324686432, 'Bangalore', '345785235', 4, '2023-07-22 09:14:11'),
(49, 'Tony', 'Stark', 'new@gmail.com', 23456781, 'Australia', '345678534', 4, '2023-07-22 09:14:11'),
(50, 'Aashma', 'brew', 'aba@gmai.com', 345678912, 'France', '8765322345', 1, '2023-07-22 09:14:11');
```

```sql
INSERT INTO `employee_credentials` (`id`, `password`, `creationTime`, `updationTime`) VALUES
(32, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-21 16:13:10', '2023-07-21 16:13:10'),
(41, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 08:27:35', '2023-07-22 08:27:35'),
(43, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 08:27:35', '2023-07-22 08:27:35'),
(44, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 09:20:03', '2023-07-22 09:20:03'),
(45, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 09:20:03', '2023-07-22 09:20:03'),
(46, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 09:20:44', '2023-07-22 09:20:44'),
(48, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 09:20:44', '2023-07-22 09:20:44'),
(49, '$2b$10$LEHD7osFjs.ewMImEGTMke80mYTRSm60wRm1MPll7wZ6JpgENTQKm', '2023-07-22 09:20:44', '2023-07-22 09:20:44');
```

Indices and Query used for indexing

Indices in any database can be used for improving the performance of search queries with where clauses. In our case the most used queries will be the netSalary attribute of the compensation table.

```
CREATE INDEX index_Net_Salary ON compensation (netSalary);
```

We can have multiple queries which would run on the employee table for id attributes as it is foreign key in multiple tables.

```
CREATE INDEX index_ID ON employee (id);
```

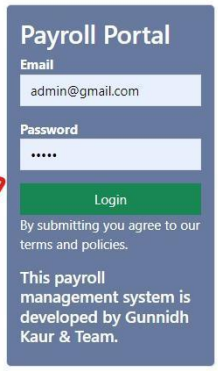
```sql
--
-- Indexes for table `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`adminId`),
  ADD UNIQUE KEY `id` (`id`);

--
-- Indexes for table `bank_details`
--
ALTER TABLE `bank_details`
  ADD PRIMARY KEY (`bankAccount`);

--
-- Indexes for table `compensation`
--
ALTER TABLE `compensation`
  ADD PRIMARY KEY (`transactionId`),
  ADD KEY `compensation_ibfk_1` (`id`),
  ADD KEY `departmentId` (`departmentId`);

--
-- Indexes for table `department`
--
ALTER TABLE `department`
  ADD PRIMARY KEY (`departmentId`),
  ADD UNIQUE KEY `department_name` (`departmentName`);

--
-- Indexes for table `employee`
--
ALTER TABLE `employee`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `email` (`email`),
  ADD UNIQUE KEY `bankAccount` (`bankAccount`),
  ADD KEY `department_id` (`departmentId`);

--
-- Indexes for table `employee_credentials`
--
ALTER TABLE `employee_credentials`
  ADD PRIMARY KEY (`id`);
```

Query used for auto increment

```sql
--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `compensation`
--
ALTER TABLE `compensation`
  MODIFY `transactionId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;

--
-- AUTO_INCREMENT for table `department`
--
ALTER TABLE `department`
  MODIFY `departmentId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `employee`
--
ALTER TABLE `employee`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=51;

--
-- AUTO_INCREMENT for table `employee_credentials`
--
ALTER TABLE `employee_credentials`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=50;
```

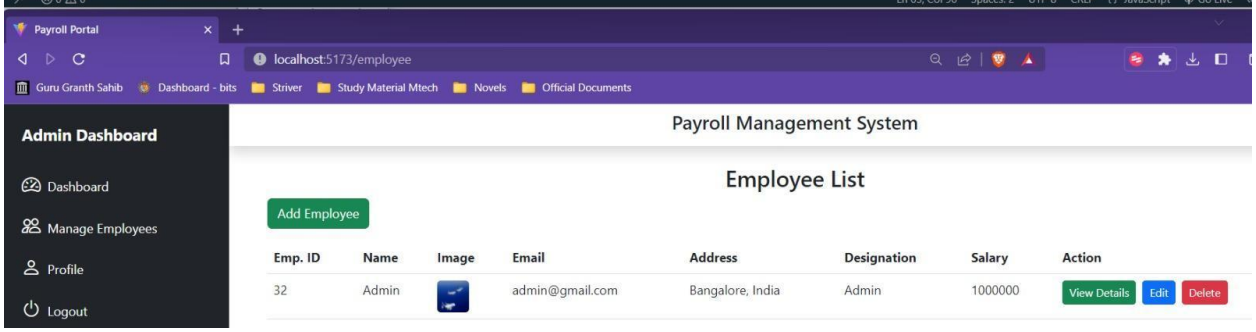
Query for logging in into the portal

```node
app.post("/login", (req, res) => {
  console.log("/login");
  const sql =
    `SELECT employee.id,
    employee_credentials.password
    FROM employee
    INNER JOIN admin ON employee.id = admin.id
    INNER JOIN employee_credentials ON employee.id =
    employee_credentials.id
    WHERE employee.email = ?`;
  con.query(sql, [req.body.email], (err, result) => {
    if (err) {
      return res.json({ Status: "Error", Error: "Error in runnig
      query" });
    }
    if (result.length > 0) {
      bcrypt.compare(
        req.body.password.toString(),
        result[0].password,
```

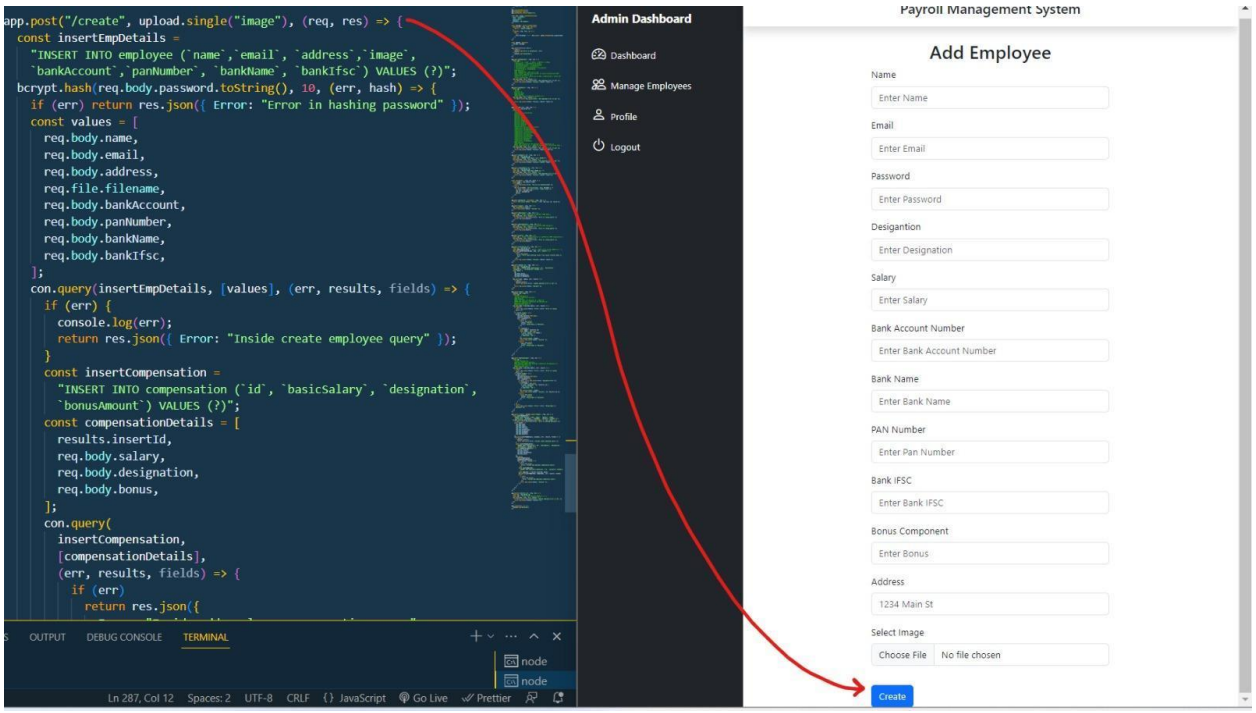


Query for getting the latest salary and designation of the employee

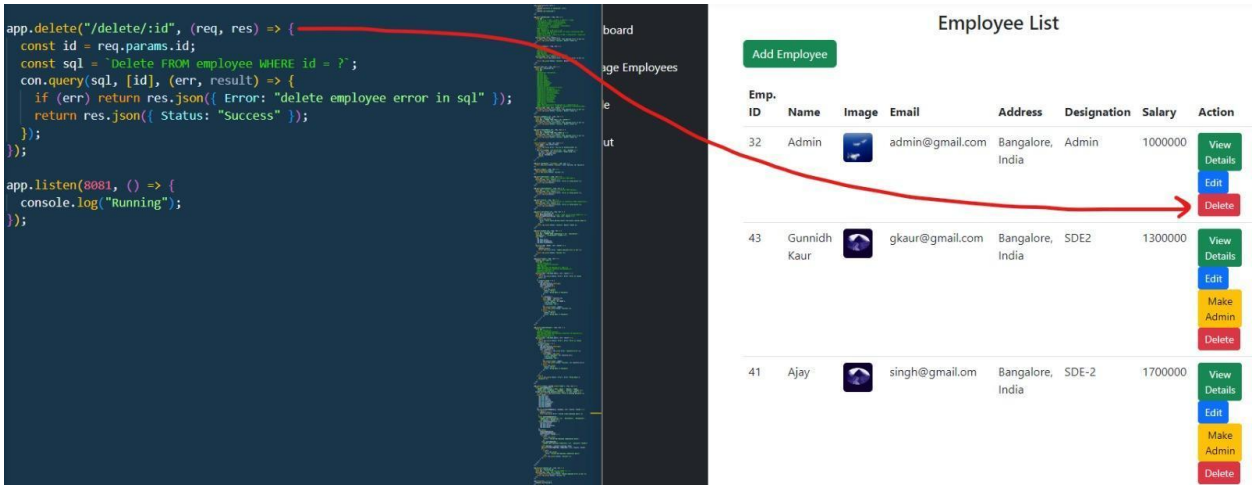
```
});

app.get("/getEmployee", (req, res) => {
  const sql =
    `SELECT e.id, e.name , e.email, e.address, e.image,
    c.designation, c.basicSalary AS salary,
    c.houseRentAllowance, c.travelAllowance,
    c.dearnessAllowance, c.grossSalary,
    c.providentFund, c.pensionFund, c.bonusAmount,
    c.creationTime as rollOutMonth
    FROM employee e
    JOIN compensation c ON e.id = c.id
    JOIN ( SELECT id, MAX(creationTime) AS latest_creationTime FROM compensation GROUP BY id )
    latest_sal ON c.id = latest_sal.id AND c.creationTime = latest_sal.latest_creationTime; ;
  con.query(sql, (err, result) => {
    if (err) return res.json({ Error: "Get employee error in sql" });
    return res.json({ Status: "Success", Result: result });
  };
});
```



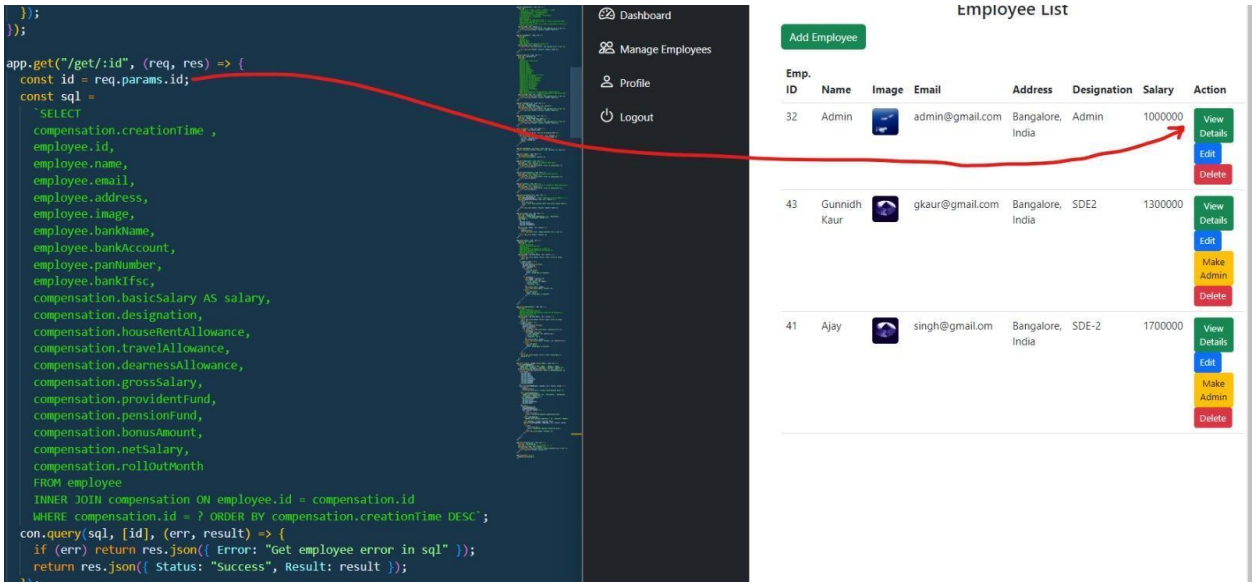
Query for creating a new employee to the table



Query for deleting the employee



Query for viewing employee details



Query to view pay slip of the employee

```
app.get("/get/:id", (req, res) => {
  const id = req.params.id;
  const sql =
    `SELECT
    compensation.creationTime ,
    employee.id,
    employee.name,
    employee.email,
    employee.address,
    employee.image,
    employee.bankName,
    employee.bankAccount,
    employee.panNumber,
    employee.bankIfsc,
    compensation.basicSalary AS salary,
    compensation.designation,
    compensation.houseRentAllowance,
    compensation.travelAllowance,
    compensation.dearnessAllowance,
    compensation.grossSalary,
    compensation.providentFund,
    compensation.pensionFund,
    compensation.bonusAmount,
    compensation.netSalary,
    compensation.rollOutMonth
    FROM employee
    INNER JOIN compensation ON employee.id = compensation.id
    WHERE compensation.id = ? ORDER BY compensation.creationTime DESC`;
  con.query(sql, [id], (err, result) => {
    if (err) return res.json({ Error: "Get employee error in sql" });
    return res.json({ Status: "Success", Result: result });
  });
});

app.get("/addAdmin/:id", (req, res) => {
```

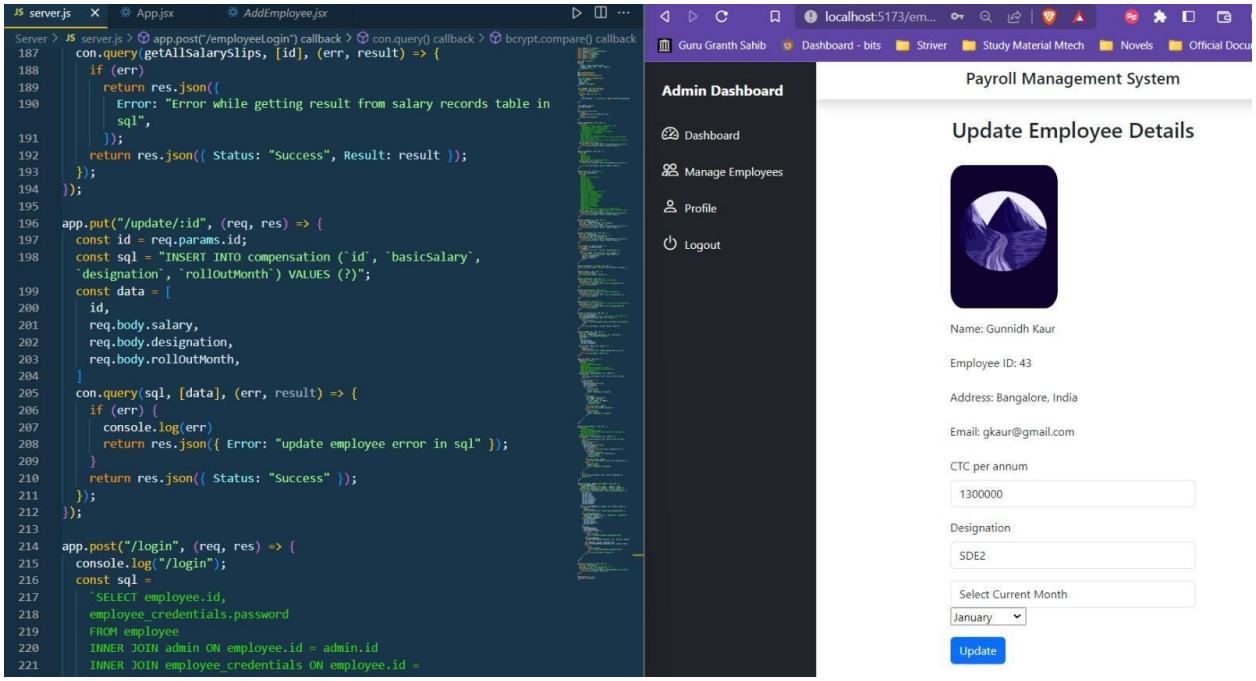
Ln 287, Col 12 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier

Employee ID:	32
Name:	Admin
Email:	admin@gmail.com
Address:	Bangalore, India
Designation:	Admin
Bank Account:	2147483647
Bank Name:	Axis Bank
Bank IFSC:	AXIS009
PAN Number:	HQ435GH89
Bonus Amount per annum:	₹ 50000
Salary per annum:	₹ 1000000

Pay Slips

DML Queries

Query for updating the employee details



Sample queries:

Show max , min and avg salary with department name for each department:

Query

Server: 127.0.0.1 » Database: db--company » Table: compensation

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

Run SQL query/queries on table db--company.compensation:

```
SELECT departmentName , MIN( netSalary ) , MAX( netSalary ) , AVG( netSalary )
from compensation as c , department as d
group BY d.departmentId;
```

Output

departmentName	MIN(netSalary)	MAX(netSalary)	AVG(netSalary)
Sales	862500	1868750	1229700.0000
Accounts	862500	1868750	1229700.0000
Engineering	862500	1868750	1229700.0000
HR	862500	1868750	1229700.0000

Show Firstname and Lastname of all employees who are admin

Query and output

```
SELECT firstName , LastName From Employee as e JOIN admin as a where e.id = a.id;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

firstName	LastName
Admin	Root

Show all | Number of rows: 25 Filter rows: Search this table

Features

- It is simple to use.
- It is completely risk-free.
- It is completely within admin's control.
- It is really engaging and saves time.
- Paperwork will be decreased.
- The computations are accurate because they are automated.
- Admin can quickly access all records whenever they are needed.

Future Scope

- Fault tolerance and Disaster Recovery scope can be included.
- This project can be migrated to cloud and can expand the scalability.
- Backup Mechanism can also be introduced.
- The system can be designed in such a way that existing functionality can be upgraded to superior versions.

Key challenges

- It requires at least one user to be there in the database who acts as an admin .
- System should comply with relevant labor laws, taxation regulations, and data protection requirements related to bank account credentials, PAN card details etc.
- It requires the internet to work.
- The project is completed within the desired timeframe, considering the complexity of the system and available resources.
- At present there is no work around for disaster recovery.

Conclusion

This project was designed with the goal of being used by both the user and the administrator. It is intended for use in small organizations with a modest number of employees. According to the requirements, the administrator can add, change, edit, and delete all employee data in his organization.
The user can access their deductions, salary components, and payslip.
The administrator can rapidly check the required records at any time. The wage of the employee is paid on a monthly basis. Several validations would allow the administrator to enter trustworthy data.

Bibliography

Websites

- www.w3schools.com
- www.tutorialspoint.com
- www.youtube.com