CS631 Term Project Deliverable (Project Report)

Best-P Company

Instructor: Vincent Oria

Team Members: Aarav Patel, Parth Patel

Submitted: May 2, 2022

Table of Contents

1. Overview

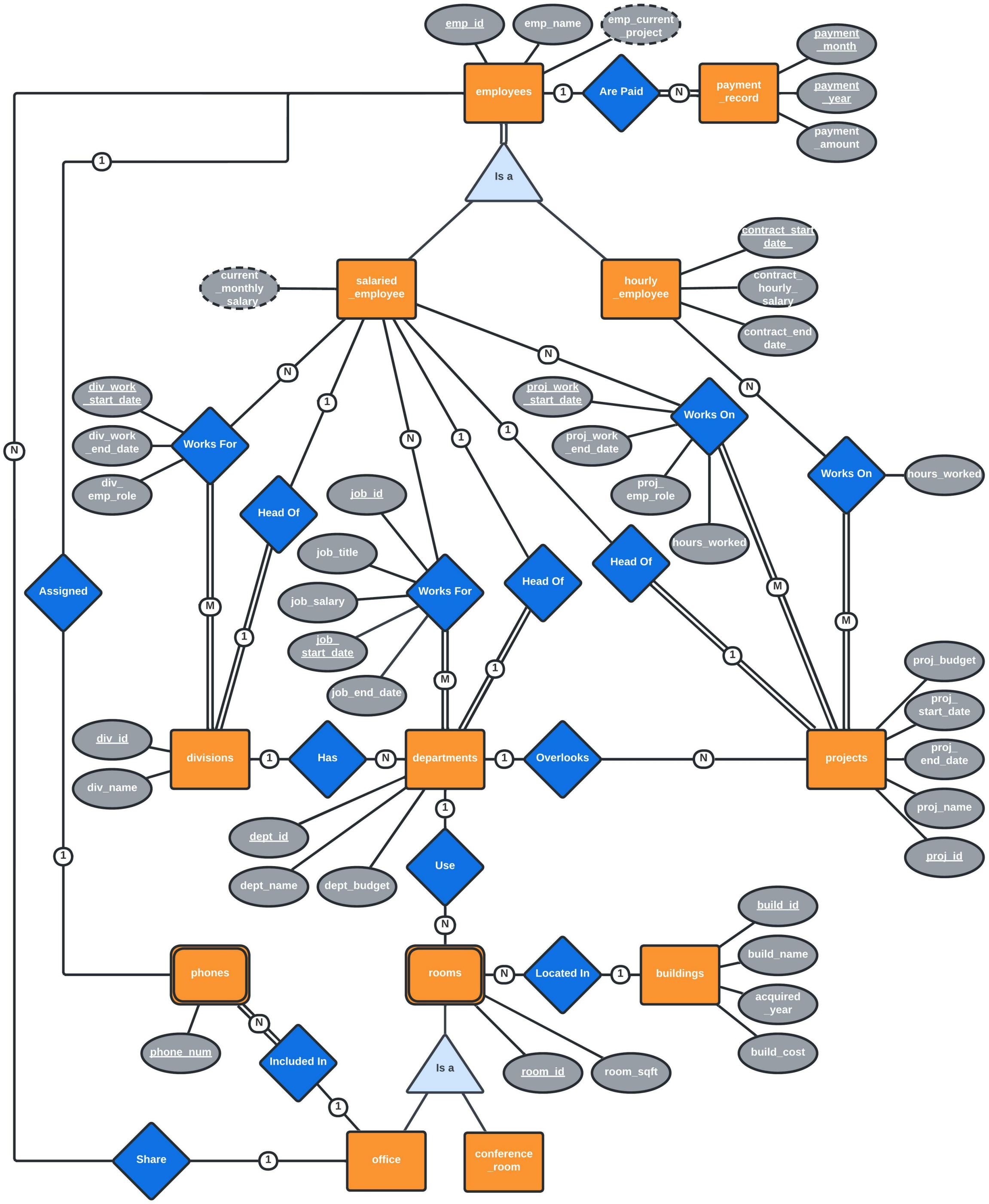
The goal of this project was to design and implement two applications for the company Best-P. The first would be one that would facilitate the HR department in managing the payroll for all of the employees of the company and pay their salaries along with keeping a record of every time an employee was paid.

The second application would aid project management. This application would assist in creating a new project, adding employees to a project, and retrieving information about projects such as project statistics and the progress of the project based on completion of project milestones.

This was achieved by implementing a small group of technologies. The database was programmed in SQL with the help of the DBMS software MySQL (through DBeaver). A mix of HTML and ThymeLeaf was used for the user interface so that the user could interact with the database by submitting or retrieving information from the database.

2. Database Design

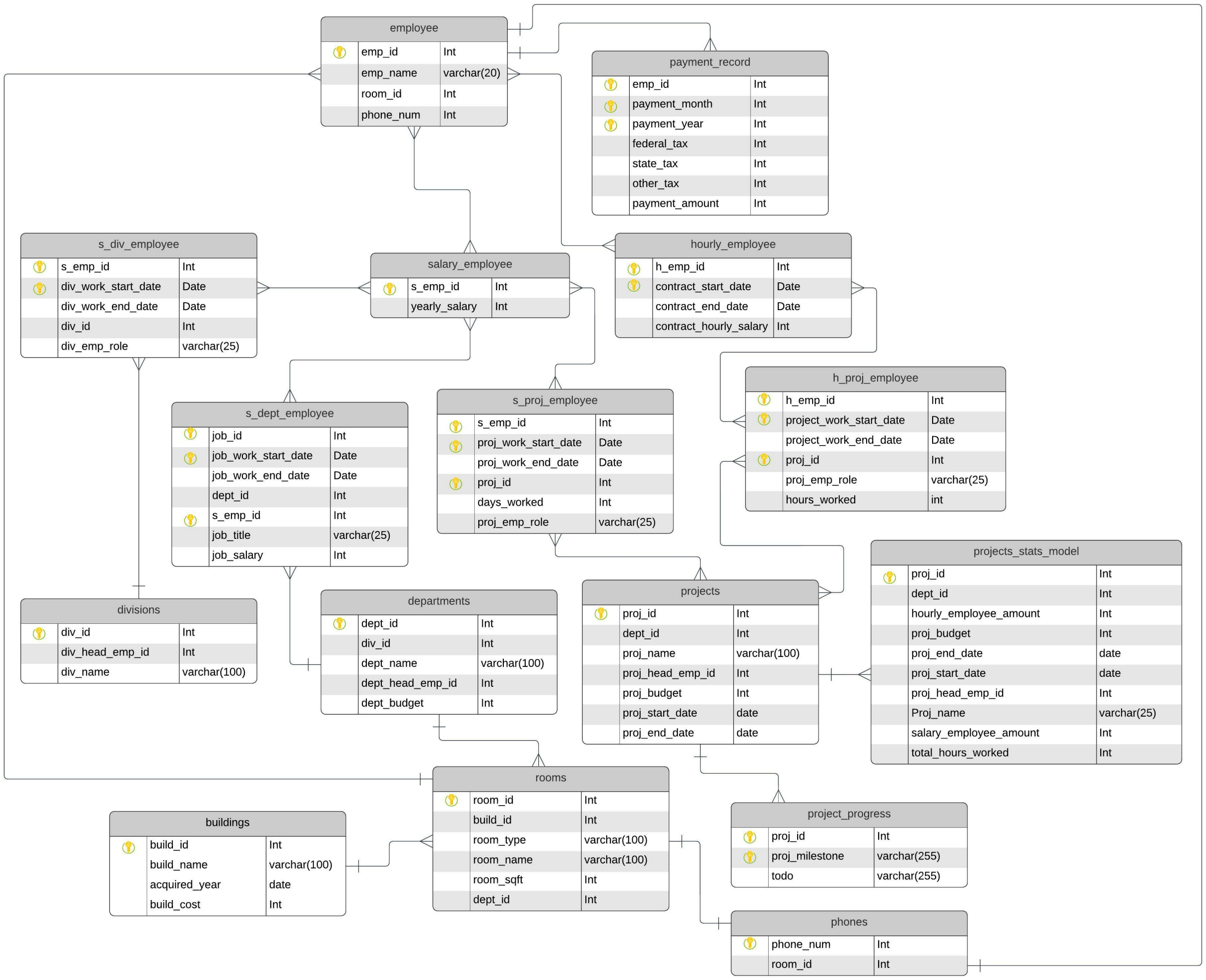
2.1 Entity-Relationship Diagram



2.1.1 Design Decisions

The ER Diagram was made in accordance with the rules laid out by the requirements. This was supplemented by feedback from the instructor to produce a final diagram on which to base the design of the database.

2.2 Relational Logical Database Design



	employee	
🔑	emp_id	Int
	emp_name	varchar(20)
	room_id	Int
	phone_num	Int

	payment_record	
🔑	emp_id	Int
🔑	payment_month	Int
🔑	payment_year	Int
	federal_tax	Int
	state_tax	Int
	other_tax	Int
	payment_amount	Int

	s_div_employee	
🔑	s_emp_id	Int
🔑	div_work_start_date	Date
	div_work_end_date	Date
	div_id	Int
	div_emp_role	varchar(25)

	salary_employee	
🔑	s_emp_id	Int
	yearly_salary	Int

	hourly_employee	
🔑	h_emp_id	Int
🔑	contract_start_date	Date
	contract_end_date	Date
	contract_hourly_salary	Int

	h_proj_employee	
🔑	h_emp_id	Int
🔑	project_work_start_date	Date
	project_work_end_date	Date
🔑	proj_id	Int
	proj_emp_role	varchar(25)
	hours_worked	int

	s_dept_employee	
🔑	job_id	Int
🔑	job_work_start_date	Date
	job_work_end_date	Date
	dept_id	Int
🔑	s_emp_id	Int
	job_title	varchar(25)
	job_salary	Int

	s_proj_employee	
🔑	s_emp_id	Int
🔑	proj_work_start_date	Date
	proj_work_end_date	Date
🔑	proj_id	Int
	days_worked	Int
	proj_emp_role	varchar(25)

	projects_stats_model	
🔑	proj_id	Int
	dept_id	Int
	hourly_employee_amount	Int
	proj_budget	Int
	proj_end_date	date
	proj_start_date	date
	proj_head_emp_id	Int
	Proj_name	varchar(25)
	salary_employee_amount	Int
	total_hours_worked	Int

	divisions	
🔑	div_id	Int
	div_head_emp_id	Int
	div_name	varchar(100)

	departments	
🔑	dept_id	Int
	div_id	Int
	dept_name	varchar(100)
	dept_head_emp_id	Int
	dept_budget	Int

	projects	
🔑	proj_id	Int
	dept_id	Int
	proj_name	varchar(100)
	proj_head_emp_id	Int
	proj_budget	Int
	proj_start_date	date
	proj_end_date	date

	rooms	
🔑	room_id	Int
	build_id	Int
	room_type	varchar(100)
	room_name	varchar(100)
	room_sqft	Int
	dept_id	Int

	buildings	
🔑	build_id	Int
	build_name	varchar(100)
	acquired_year	date
	build_cost	Int

	project_progress	
🔑	proj_id	Int
🔑	proj_milestone	varchar(255)
	todo	varchar(255)

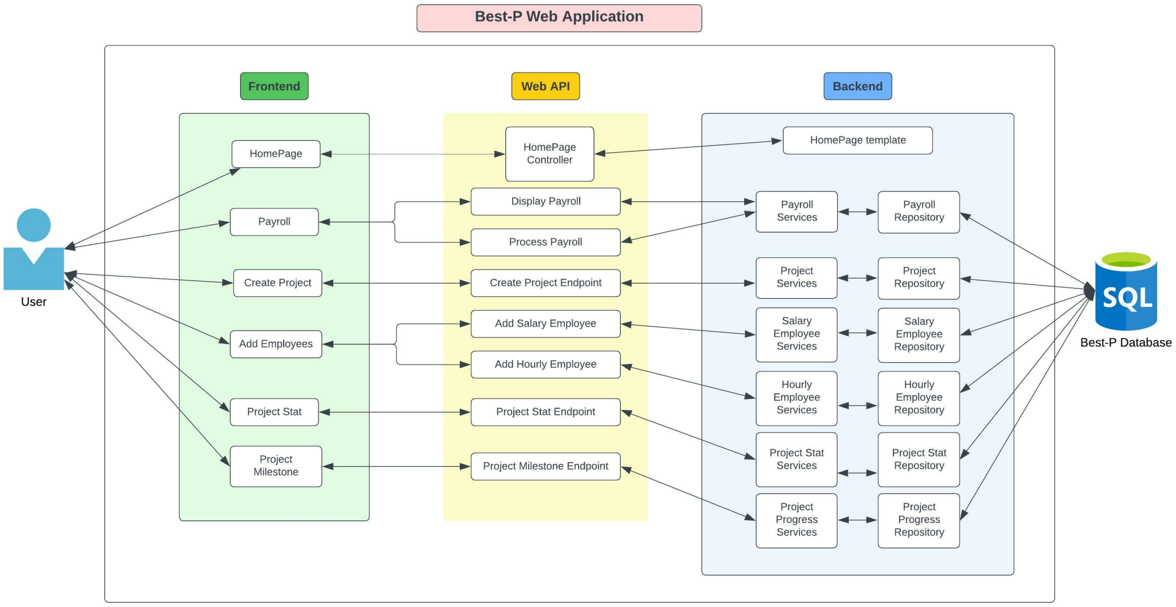
	phones	
🔑	phone_num	Int
	room_id	Int

2.2.1 Design Decisions

The database reflects all of the entities and relations described in the ER diagram provided in section 2.1. One change that was made to facilitate the retrieval of the project statistics was the addition of a project stats table. This extra table has no additional bearing on the relational database or the application at large.

3. Application Design

3.1 Application Program Design

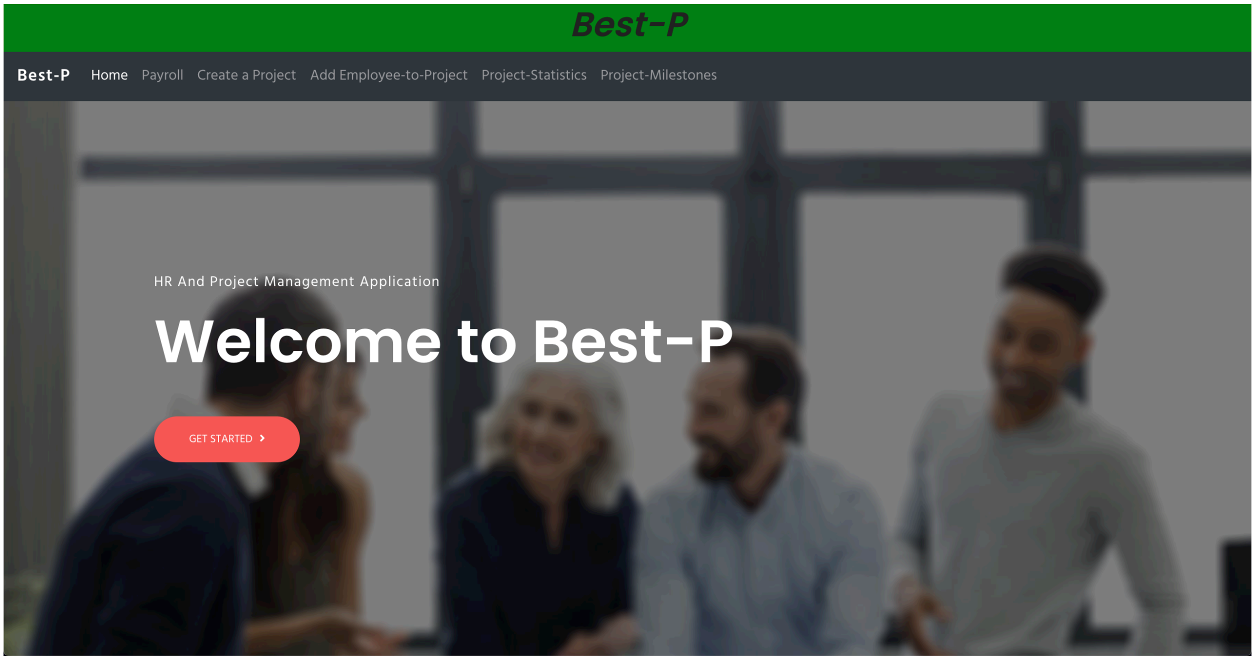


3.1.1 Design Decisions

We decided to use Model-View-Controller (MVC) software design architecture for this application to create and maintain stability of the application. The three interconnected elements would help us to develop a robust application for the user.

4. User Manual

4.1 Home Page



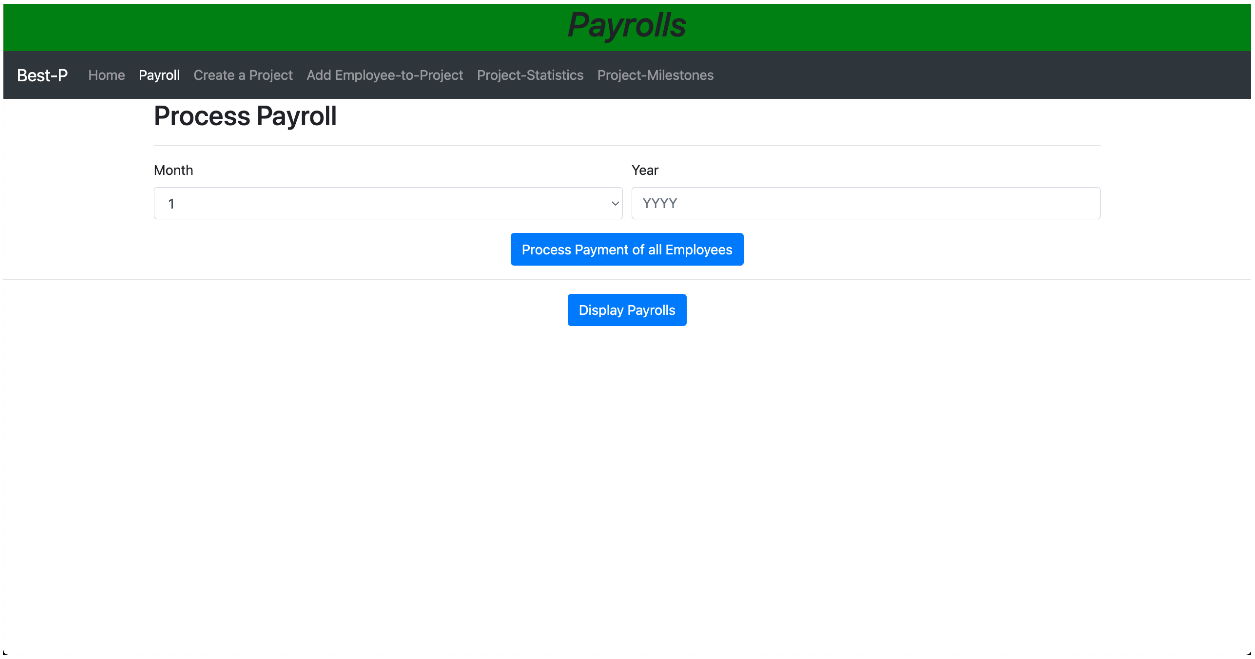
Best-P

HR And Project Management Application

Welcome to Best-P

Usage: This page is used for welcoming the user and providing access to all of the features of the Project such as payroll process, creating a project, adding employees to project, project statistics, and project milestones.

4.2 Payroll



Month: 1

Year: YYYY

This page will be used for processing the payroll and retrieving salary/wage payment history.

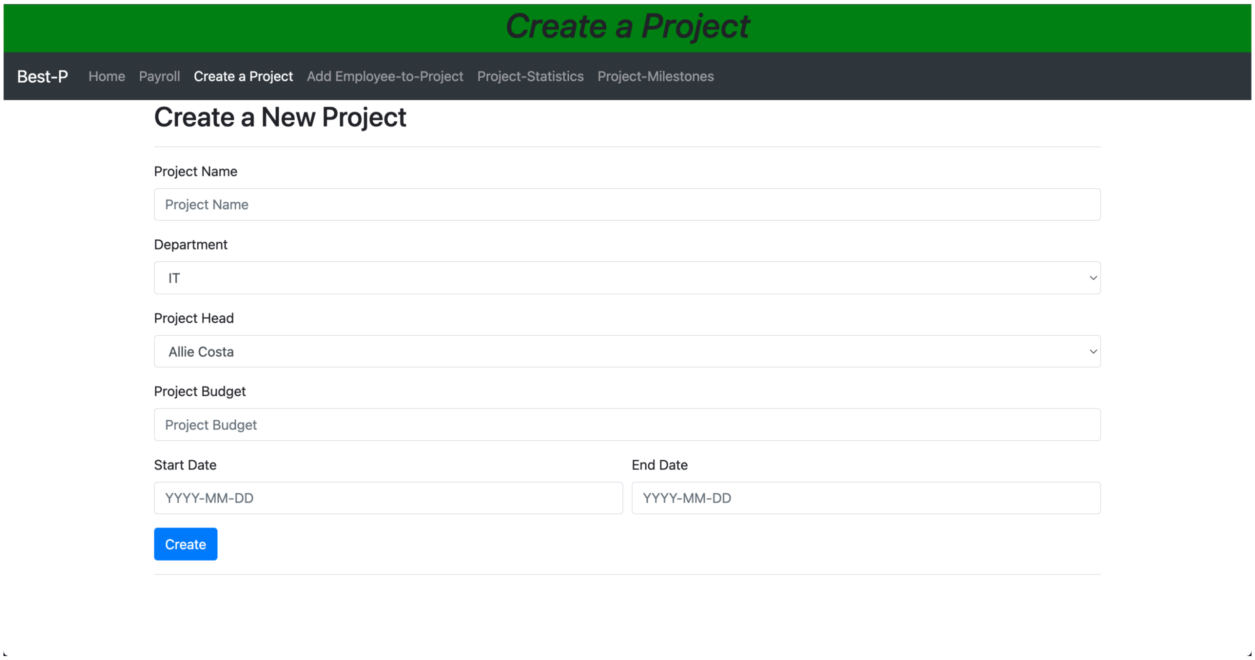
4.2.1 Process Employee Payroll

Here you will be able to pay the salaries/wages for all of the employees working for the company for any given month and year of your choice. Once you have put in your selections, click the button "Process Payment of all Employees". This will generate payment records for all of the employees and show you the generated records. The records will show the Gross Payment made, the federal tax, state tax, and other tax paid, for each employee for that given month and year.

4.2.2 Display Payroll

This functionality will allow you to view the payment history/payroll for all of the employees. Clicking the "Display Payrolls" button will show you all of the records of payment to each employee for each instance of payment.

4.3 Create a Project



Create a New Project

Project Name
Project Name

Department
IT

Project Head
Allie Costa

Project Budget
Project Budget

Start Date
YYYY-MM-DD

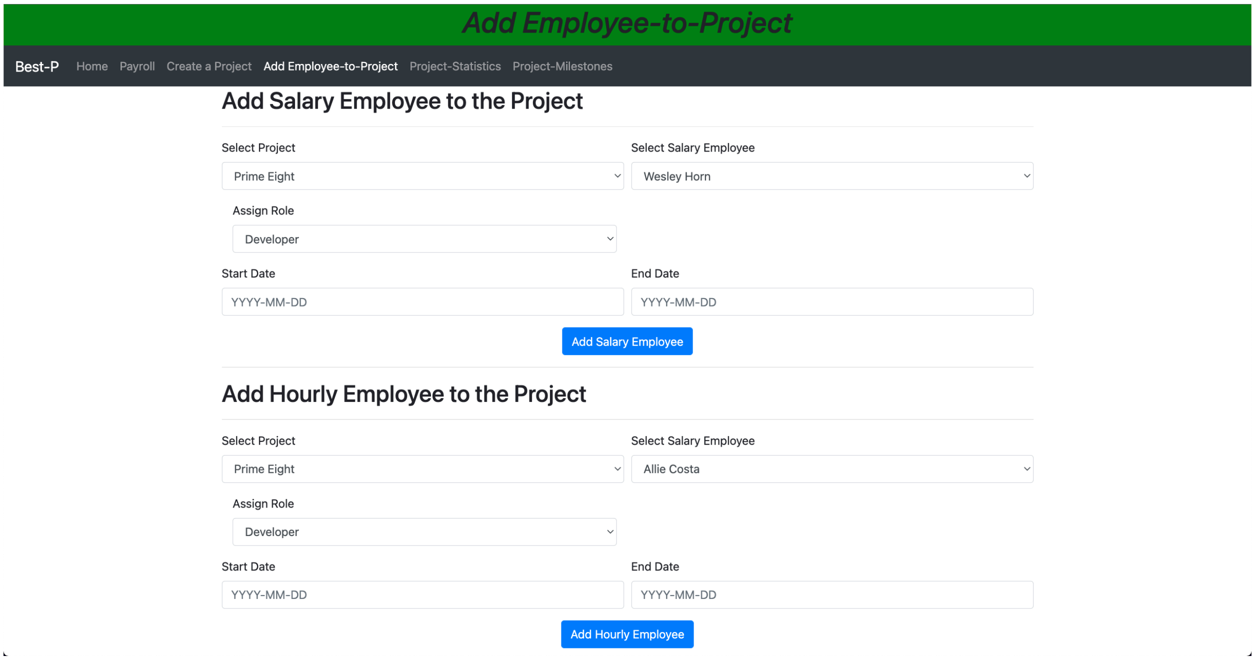
End Date
YYYY-MM-DD

This page allows you to create a new project for the company. You will be prompted to input the following information:

- **Project Name**: The name of the new project
- **Department**: Which department this project will be overlooked by
- **Project Head**: Which employee will head the project
- **Project Budget**: What the budget of the project will be
- **Start Date**: The start date of the project
- **End Date**: The end date of the project

After having filled out those fields, click on the "Create" button. Your project will have been created and you will be able to add employees to that project in the next tab.

4.4 Add Employees to a Project



Add Salary Employee to the Project

Select Project: Prime Eight

Select Salary Employee: Wesley Horn

Assign Role: Developer

Start Date: YYYY-MM-DD

End Date: YYYY-MM-DD

Add Hourly Employee to the Project

Select Project: Prime Eight

Select Salary Employee: Allie Costa

Assign Role: Developer

Start Date: YYYY-MM-DD

End Date: YYYY-MM-DD

This page will allow you to add employees to the project who work on a salaried or hourly basis.

4.4.1 Add Salary Employee

Here you can add an employee who works for the company for a salary. You will be prompted to fill out the following fields:

- **Project**: The project that you want to add the employee to
- **Salary Employee**: The salaried employee that you want to add to the project
- **Role**: The role the employee will play in the project
- **Start Date**: The start date of the employee working on the project
- **End Date**: The end date of the employee working on the project

After you have made your selections and input the values you'd like, press the "Add Salary Employee" button. This will add the salaried employee of your choice to the project of your choice.

4.4.2 Add Hourly Employee

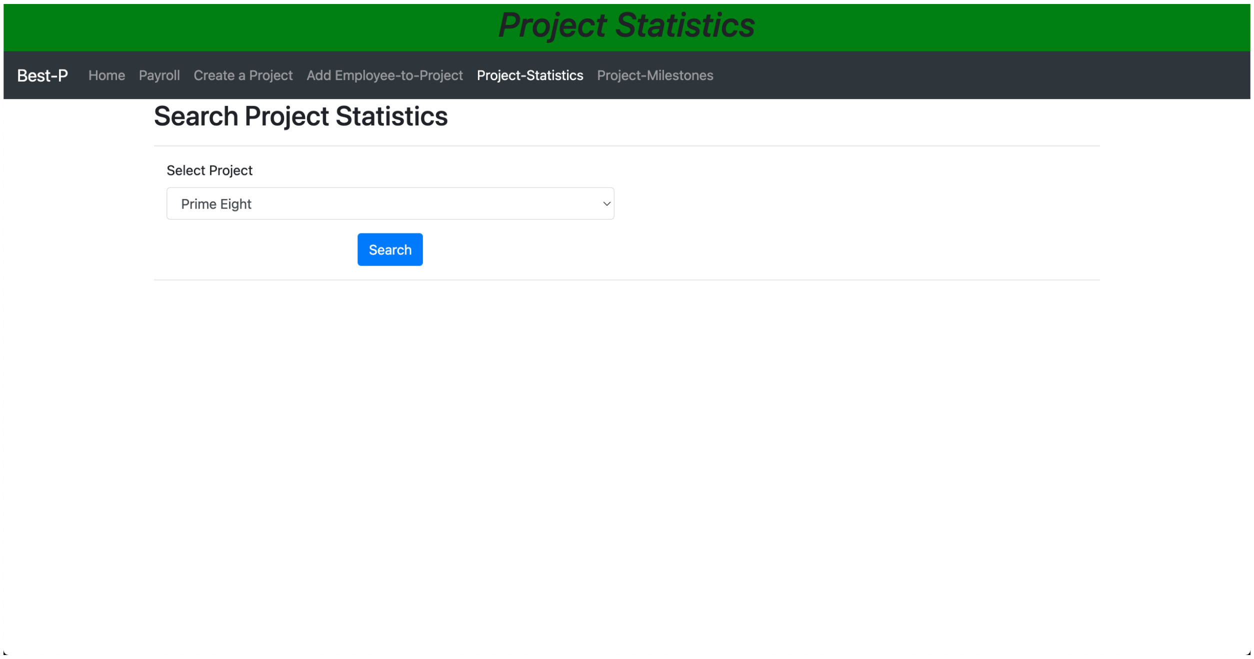
Here you can add an employee who works for the company on an hourly basis (hired for a project). You will be prompted to fill out the following fields:

- **Project**: The project that you want to add the employee to
- **Hourly Employee**: The hourly employee that you want to add to the project
- **Role**: The role the employee will play in the project

- **Start Date**: The start date of the employee working on the project
- **End Date**: The end date of the employee working on the project

After you have made your selections and input the values you'd like, press the "Add Hourly Employee" button. This will add the salaried employee of your choice to the project of your choice.

4.5 Project Statistics



Search Project Statistics

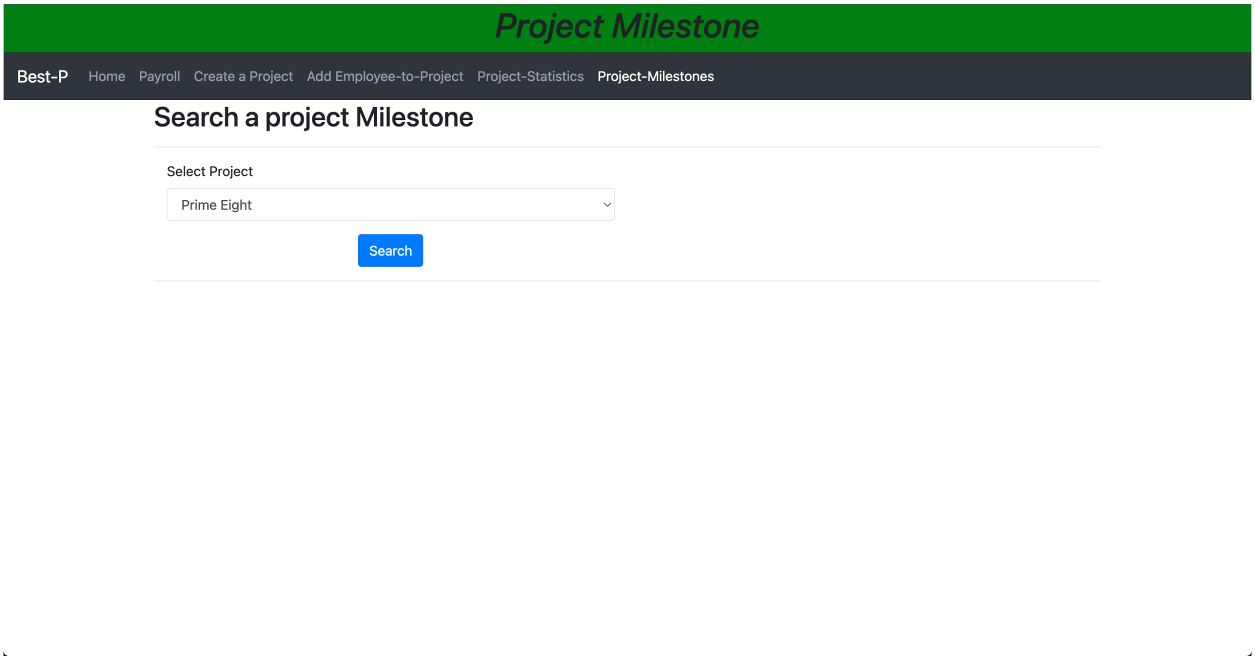
Select Project

Prime Eight

This page allows you to view the following statistics for any project that you select from a dropdown list.

- The Project ID
- Name of the project
- The total number of hours worked by all of the employees on the project
- The total number of salaried employees
- The total number of hourly employees

4.6 Project Milestones



Project Milestone

Search a project Milestone

Select Project: Prime Eight

This page allows you to view the progress of any given project based on the milestones that have been set and whether or not that has been completed/ achieved. You can select whichever project you want to view the progress of with the dropdown list provided and then clicking the “Search” button.

5. Assumptions / General Design Decisions

The assumptions that were made for the project are listed below:

- We assumed that there were two types of employees that worked for the company. The first was full-time employees that were given a salary (referred to as salaried employees in our project). The second was employees who were hired for a project and were being paid hourly (referred to as hourly employees in our project). Though the distinction was made, any further complication that arises from such a distinction was not considered, and a more simplistic approach was taken wherever necessary.
- We assumed that project milestones are not time bound, but are rather to be seen as a set of steps necessary to see the project to completion. Hence there are no dates, just whether the milestone was achieved or not.
- Because the project requirements did not say to provide the functionality to create/hire employees (salaried or hourly), the functionalities that required adding employees to a project were implemented by allowing the user to select an employee from an existing pool of employees already in the database. This means that though hourly employees are hired just for a project, the assumption here is that we are selecting an hourly employee from an existing list of hirable employees and both hiring them and assigning them to a project at the same time.
- We assumed that all employees (both salaried and hourly) would be paid at the same time every month. Hence we are allowing the user to select which month and input which year they would like to pay all of the employees for. Due to the vague wording of the requirement and obvious time constraints, no effort was made to automate this functionality. Furthermore, automating it would not have allowed us to demonstrate it working.
- Though a lot of the other aspects of the company were laid out in the requirements (such as phone numbers, rooms, buildings, etc.), no effort has been made to display such data due to the functionality of the two applications described in the requirements not requiring us to do so. Hence, no functionality has been provided for viewing the details of any single employee, room, building, etc. or any combination of them. The tables have been created, as required by the implementation of the ER diagram. However, some of them are not accessed/used for any of the functionalities provided. That being the case, the tables not used have not been populated either as we saw no need for that.
- Extra attributes were added to entities on top of the ones listed out in the requirements to provide more clarity. (Such as to Division and Project so that they had names, etc.)

- We also opted to have the functionalities of the two proposed applications combined into one “website” with separate tabs for each aspect that was required under that requirement section. This was so that the website was a “one-stop-shop” for everything that would need to be done based on the requirements. We have placed the tabs for the HR management application first, and then provided the tabs for the Project management application. To split them into two separate applications would’ve required double the amount of effort, and, keeping time constraints in mind, we assumed this would be an understandable compromise solution that still provided for all of the requirements.

Appendix: SQL (For generating database and tables, and populating the database.)

The following SQL was created in accordance with the database tool we used. Comments have been added to provide clarity and to group the commands.

```
-- Creating tables for database

CREATE TABLE `departments` (
  `dept_id` int(11) NOT NULL,
  `dept_budget` int(11) DEFAULT NULL,
  `dept_head_emp_id` int(11) DEFAULT NULL,
  `dept_name` varchar(255) DEFAULT NULL,
  `div_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `divisions` (
  `div_id` int(11) NOT NULL,
  `div_head_emp_id` int(11) DEFAULT NULL,
  `div_name` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `buildings` (
  `build_id` int(11) NOT NULL,
  `build_acquired_year` int(11) DEFAULT NULL,
  `build_cost` int(11) DEFAULT NULL,
  `build_name` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `employees` (
  `emp_id` int(11) NOT NULL,
  `emp_name` varchar(255) DEFAULT NULL,
  `phone_num` int(11) DEFAULT NULL,
  `room_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Spring Boot generated table (ignore)
CREATE TABLE `hibernate_sequence` (
  `next_val` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `hourly_employee` (
  `contract_start_date` varchar(255) NOT NULL,
  `h_emp_id` int(11) NOT NULL,
  `contract_end_date` varchar(255) DEFAULT NULL,
  `contract_hourly_salary` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `h_proj_employee` (
  `h_emp_id` int(11) NOT NULL,
  `proj_work_start_date` varchar(255) NOT NULL,
  `hours_worked` int(11) DEFAULT NULL,
  `proj_emp_role` varchar(255) DEFAULT NULL,
  `proj_id` int(11) NOT NULL,
  `proj_work_end_date` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `payment_record` (
  `emp_id` int(11) NOT NULL,
  `payment_month` int(11) NOT NULL,
  `payment_year` int(11) NOT NULL,
  `federal_tax` int(11) DEFAULT NULL,
  `other_tax` int(11) DEFAULT NULL,
  `payment_amount` int(11) DEFAULT NULL,
  `state_tax` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `phones` (
  `phone_num` int(11) NOT NULL,
  `room_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `projects` (
  `proj_id` int(11) NOT NULL,
  `dept_id` int(11) DEFAULT NULL,
  `proj_budget` int(11) DEFAULT NULL,
  `proj_end_date` varchar(255) DEFAULT NULL,
  `proj_head_emp_id` int(11) DEFAULT NULL,
  `proj_name` varchar(255) DEFAULT NULL,
  `proj_start_date` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `project_progress` (
  `proj_id` int(11) NOT NULL,
  `proj_milestone` varchar(255) NOT NULL,
  `Completed` tinyint(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `project_stats_model` (
  `proj_id` int(11) NOT NULL,
  `dept_id` int(11) NOT NULL,
  `hourly_employee_amount` int(11) NOT NULL,
  `proj_budget` int(11) NOT NULL,
  `proj_end_date` varchar(255) DEFAULT NULL,
  `proj_head_emp_id` int(11) NOT NULL,
  `proj_name` varchar(255) DEFAULT NULL,
  `proj_start_date` varchar(255) DEFAULT NULL,
  `salary_employee_amount` int(11) NOT NULL,
  `total_hours_worked` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `rooms` (
  `room_id` int(11) NOT NULL,
  `build_id` int(11) DEFAULT NULL,
  `dept_id` int(11) DEFAULT NULL,
  `room_sqft` int(11) DEFAULT NULL,
  `room_type` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `salary_employee` (
  `s_emp_id` int(11) NOT NULL,
  `yearly_salary` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `s_dept_employee` (
  `job_id` int(11) NOT NULL,
  `job_start_date` varchar(255) NOT NULL,
  `s_emp_id` int(11) NOT NULL,
  `dept_id` int(11) DEFAULT NULL,
  `job_end_date` varchar(255) DEFAULT NULL,
  `job_salary` int(11) DEFAULT NULL,
  `job_title` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `s_div_employee` (
  `div_work_start_date` varchar(255) NOT NULL,
  `s_emp_id` int(11) NOT NULL,
  `div_emp_role` varchar(255) DEFAULT NULL,
  `div_id` int(11) DEFAULT NULL,
  `div_work_end_date` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `s_proj_employee` (
  `proj_id` int(11) NOT NULL,
  `proj_work_start_date` varchar(255) NOT NULL,
  `s_emp_id` int(11) NOT NULL,
  `days_worked` int(11) DEFAULT NULL,
  `proj_emp_role` varchar(255) DEFAULT NULL,
  `proj_work_end_date` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;




-- Insert statements for populating the databse

INSERT INTO `departments` (`dept_id`, `dept_budget`,
`dept_head_emp_id`, `dept_name`, `div_id`) VALUES
(456, 100000, 159, 'Cyber', 303),
(4488, 10000, 784, 'IT', 125),
(7894, 100000, 9564, 'Accountings', 5456);

INSERT INTO `employees` (`emp_id`, `emp_name`, `phone_num`,
`room_id`) VALUES
(159, 'Allie Costa', 798859000, 325),
(523, 'Wesley Horn', 879846613, 85),
(741, 'Jeremy Curry', 646416113, 64),
(963, 'Jose Wells', 646316764, 15),
(4560, 'Riya Bell', 654646510, 5454),
(6345, 'Eliza Roman', 798859546, 4577);

-- Spring Boot generated table (ignore)
INSERT INTO `hibernate_sequence` (`next_val`) VALUES
(5);

INSERT INTO `hourly_employee` (`contract_start_date`,
`h_emp_id`, `contract_end_date`, `contract_hourly_salary`)
VALUES
('2022-01-05', 159, '2029-01-10', 30),
('2022-02-04', 963, '2025-07-05', 25);

INSERT INTO `h_proj_employee` (`h_emp_id`,
`proj_work_start_date`, `hours_worked`, `proj_emp_role`,
`proj_id`, `proj_work_end_date`) VALUES
(963, '2022-01-06', 640, 'Analyst', 2, '2022-04-06'),
(963, '2022-02-04', 480, 'Front-end Developer', 1234,
'2023-02-06'),
(963, '2022-04-01', 160, 'UI-Designer', 1, '2023-05-05'),
(963, '2022-05-06', 0, 'Analyst', 3, '2025-01-05'),
(963, '2022-06-20', 0, 'Analyst', 4, '2023-04-05');
```

```
INSERT INTO `payment_record` (`emp_id`, `payment_month`,
`payment_year`, `federal_tax`, `other_tax`, `payment_amount`,
`state_tax`) VALUES
(523, 1, 2026, 83, 24, 833, 41),
(741, 1, 2025, 416, 124, 4166, 208),
(741, 1, 2026, 416, 124, 4166, 208),
(963, 1, 2025, 400, 120, 4000, 200),
(963, 1, 2026, 400, 120, 4000, 200);

INSERT INTO `projects` (`proj_id`, `dept_id`, `proj_budget`,
`proj_end_date`, `proj_head_emp_id`, `proj_name`,
`proj_start_date`) VALUES
(1, 4488, 1022, '2023-06-02', 4560, 'Prime Eight',
'2022-02-06'),
(2, 4488, 15000, '2025-04-06', 4560, 'Accounting Webapp',
'2022-03-01'),
(3, 4488, 600000, '2026-05-01', 6345, 'Data Warehouse Project',
'2022-05-01'),
(4, 4488, 800000, '2026-04-01', 6345, 'Employee Portal Project',
'2022-06-01'),
(1234, 456, 1000, '2023-02-06', 741, 'Severe', '2022-02-04');

INSERT INTO `project_progress` (`proj_id`, `proj_milestone`,
`Completed`) VALUES
(1234, 'The project will start its design phase by next week',
1);

INSERT INTO `salary_employee` (`s_emp_id`, `yearly_salary`)
VALUES
(523, 10000),
(741, 50000);

INSERT INTO `s_proj_employee` (`proj_id`,
`proj_work_start_date`, `s_emp_id`, `days_worked`,
`proj_emp_role`, `proj_work_end_date`) VALUES
(1, '2022-02-04', 741, 60, 'Developer', '2023-02-06'),
(2, '2022-01-06', 741, 80, 'UI-Designer', '2022-06-01'),
(3, '2022-05-01', 741, 0, 'Developer', '2023-05-01'),
(4, '2022-05-01', 741, 0, 'UI-Designer', '2023-04-05'),
(1234, '2022-02-04', 741, 60, 'Developer', '2023-02-06');



-- Setting Primary Keys
```

```
ALTER TABLE `departments`
  ADD PRIMARY KEY (`dept_id`);

ALTER TABLE `divisions`
  ADD PRIMARY KEY (`div_id`);

ALTER TABLE `buildings`
  ADD PRIMARY KEY (`build_id`);

ALTER TABLE `employees`
  ADD PRIMARY KEY (`emp_id`);

ALTER TABLE `hourly_employee`
  ADD PRIMARY KEY (`contract_start_date`,`h_emp_id`);

ALTER TABLE `h_proj_employee`
  ADD PRIMARY KEY (`h_emp_id`,`proj_work_start_date`,`proj_id`);

ALTER TABLE `payment_record`
  ADD PRIMARY KEY (`emp_id`,`payment_month`,`payment_year`);

ALTER TABLE `phones`
  ADD PRIMARY KEY (`phone_num`);

ALTER TABLE `projects`
  ADD PRIMARY KEY (`proj_id`);

ALTER TABLE `project_progress`
  ADD PRIMARY KEY (`proj_id`,`proj_milestone`);

ALTER TABLE `project_stats_model`
  ADD PRIMARY KEY (`proj_id`);

ALTER TABLE `rooms`
  ADD PRIMARY KEY (`room_id`);

ALTER TABLE `salary_employee`
  ADD PRIMARY KEY (`s_emp_id`);

ALTER TABLE `s_dept_employee`
  ADD PRIMARY KEY (`job_id`,`job_start_date`,`s_emp_id`);

ALTER TABLE `s_div_employee`
  ADD PRIMARY KEY (`div_work_start_date`,`s_emp_id`);

ALTER TABLE `s_proj_employee`
```

```
  ADD PRIMARY KEY (`proj_id`,`proj_work_start_date`,`s_emp_id`);

-- Commit changes
COMMIT;
```