

Great Employees

Vision

“Great Employees” is web-application which allows to control amount of employees and departments.

Application should provide:

- Storing employees and departments in database(db);
- Display list of employees(emp.) and departments(dep.);
- Updating lists of emp. and dep. (adding, editing, removing);
- Display average salary for each dep.;
- Searching emp. for birth date or between birth dates;
- Sort columns in emp. table;

1. Employees

1.1 Display list of employees

The mode is designed to view the list of employees, with the exact date birth, or for period between birth dates.

Main scenario:

- User select date in ‘DATE FROM’ and push ‘Search’(if he wants to search employees with exact date birth)
- User select date in ‘DATE FROM’ and date in ‘DATE TO’ and push ‘Search’(if he wants to search employees whose birth dates between ‘DATE FROM’ and ‘DATE TO’)

MainEmployeesAdd EmployeeDepartmentsAdd Department

Search

DATE FROM

11 / 10 / 1996

DATE TO

mm / dd / yyyy

Search

Page: 1 from 10

ID	NAME	BIRTHDATE	DEPARTMENT	SALARY	ACTIONS
127	Adriane Potter	1999-06-26	Research and development	4342	Profile Delete
1005	Alina Kireeva	1998-01-05	Retouch	2000	Profile Delete
527	Alisa Dunn	2002-12-08	Research and development	2621	Profile Delete
485	Alisa Greer	2002-03-26	Accounts and Finance	4290	Profile Delete
597	Antoinette Cooke	2004-07-13	Security and transport	4137	Profile Delete
616	Antoinette Dunn	2003-11-05	Learning and development	2635	Profile Delete
84	Antoinette Hansen	2004-10-27	Product development	3664	Profile Delete
792	Antoinette Lowery	2000-04-29	Security and transport	1958	Profile Delete
265	Antoinette Orozco	1997-05-08	Accounts and Finance	529	Profile Delete
930	Antoinette Pierce	2000-04-01	Accounts and Finance	4643	Profile Delete
871	Antoinette Swanson	1997-03-12	Product development	4424	Profile Delete

Pic. 1.1 View the Employees list

The list displays the following columns:

- ID - unique employee id;
- NAME - employee's name;
- BIRTHDATE - birth date of employee;
- DEPARTMENT - department where employee works;
- SALARY - salary of employee;
- ACTIONS - possible actions for this employee;

Ordering by columns:

- In the headers of columns located arrows. User can press any of them and table will be sorted by that column in the way, that user chose.

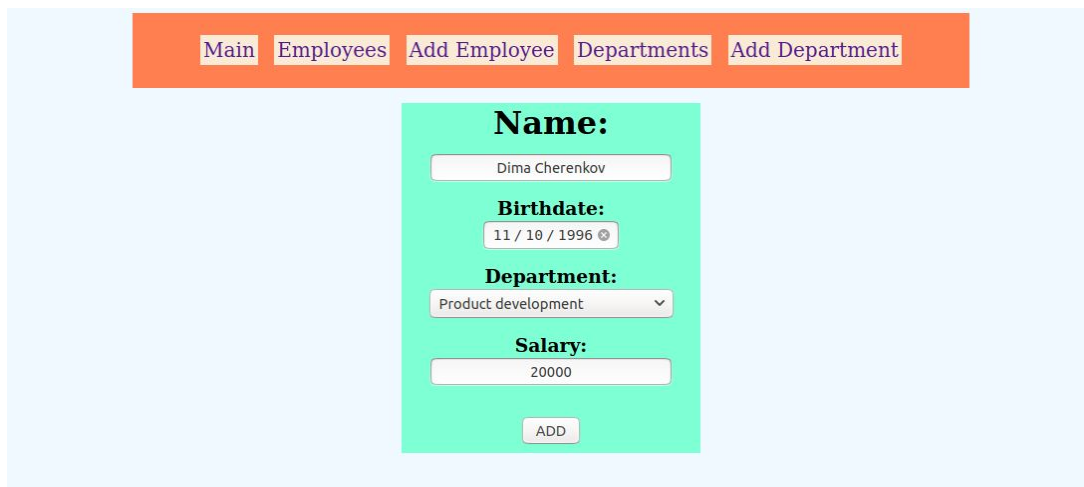
1.2 Add employee

Main scenario:

- User clicks 'Add Employee' button in the header;
- Application displays form to enter data;
- User enters data and press 'ADD' button;
- If any field are empty, error message displays;
- If all fields are field, the record is adding to db;
- If error occurs, then error message displays;

Cancel operation scenario:

- User clicks 'Add Employee' button in the header;
- Application displays form to enter data;
- User leaves page;
- Data don't save to db;



Pic. 1.2 Add employee

When adding an employee, the following details are entered:

- Employee's name;
- Employee's birth date;
- Employee's department;
- Employee's salary;

1.3 Edit employee

Main scenario:

- User clicks the 'Profile' button in the employee list;
- Application displays form with existing data;
- User enters data and presses 'CHANGE' button;
- If any field are empty, error message displays;
- If all fields are field, the record is adding to db;
- If error occurs, then error message displays;

Cancel operation scenario:

- User clicks the 'Profile' button in the employee list;
- Application displays form with existing data;
- User leaves page;
- Data don't save to db;

Main **Employees** **Add Employee** **Departments** **Add Department**

Name:
Adriane Potter

Birthdate:
1999-06-26

Department:
Accounts and Finance

Salary:
4342

[CHANGE](#) [DELETE](#)

Pic. 1.3 Edit employee

When editing an employee, the following details are entered:

- Employee's name;
- Employee's birth date;
- Employee's department;
- Employee's salary;

1.4 Removing employee

Main scenario:

- The user, while in the list of employees, presses the 'Delete' button in the 'ACTIONS' column;
- If the employee can be removed, it's deleting from db;
- If error occurs, then error message displays;
- If order record is successfully deleted, then list of employees without deleted records is displaying;

The screenshot shows a web application interface for managing employees. At the top, there is a navigation bar with links: Main, Employees, Add Employee, Departments, and Add Department. Below this, a search form is displayed with fields for 'DATE FROM' (11 / 10 / 1996) and 'DATE TO' (mm / dd / yyyy), and a 'Search' button. Below the search form, it says 'Page: 1 from 1'. A table lists employee records with columns: ID, NAME, BIRTHDATE, DEPARTMENT, SALARY, and ACTIONS. The table contains one record for Dima Cherenkov. Below the table, it says 'Pages: Previous 1 ... 1 Next'.

ID	NAME ↑ ↓	BIRTHDATE ↑ ↓	DEPARTMENT ↑ ↓	SALARY ↑ ↓	ACTIONS
1004	Dima Cherenkov	1996-11-10	Product development	20002	Profile Delete

Pic. 1.4 Deleting employee(1 var.)

Cancel operation scenario:

- Cancel operation scenario for this function doesn't exist;

Side scenario:

- The user, while in the editing employee form, presses the 'DELETE' button next to after 'CHANGE' button;
- If the employee can be removed, it's deleting from db;
- If error occurs, then error message displays;
- If employee record is successfully deleted, then search form for employees will be shown;

The screenshot shows the 'Edit Employee' form. It has a navigation bar with links: Main, Employees, Add Employee, Departments, and Add Department. The form fields are: Name (Alina Kireeva), Birthdate (1998-01-05), Department (Retouch), and Salary (2000). At the bottom, there are two buttons: 'CHANGE' and 'DELETE'.

Pic. 1.5 Deleting employee(2 var.)

Cancel operation scenario:

- Cancel operation scenario for this function doesn't exist;

2. Departments

2.1 Display list of departments

The mode is designed to view the list of departments, with the average salary.

Main scenario:

- User select 'Departments' in header;

Main Employees Add Employee Departments Add Department			
id	DEPARTMENT	AVERAGE SALARY	ACTIONS
16	Accounts and Finance	2756.5625	Delete Change
9	Admin department	2624.3814	Delete Change
2	HR	2944.6635	Delete Change
4	Infrastructures	2873.2824	Delete Change
7	IT services	2765.8415	Delete Change
6	Learning and development	2817.6061	Delete Change
8	Product development	2979.7565	Delete Change
5	Research and development	2723.7030	Delete Change
13	Retouch	2000.0000	Delete Change
3	Sales and marketing (Business development)	2914.4356	Delete Change
10	Security and transport	2813.4476	Delete Change

Pic. 2.1 View the Departments list

The list displays the following columns:

- ID - unique department id;

- DEPARTMENT - department's name;
- AVERAGE SALARY - average salary for employees in this department;
- ACTIONS - possible actions for this department;

1.2 Add department

Main scenario:

- User clicks 'Add Department' button in the header;
- Application displays form to enter data;
- User enters data and press 'ADD' button;
- If any field are empty, error message displays;
- If all fields are field, the record is adding to db;
- If error occurs, then error message displays;

Cancel operation scenario:

- User clicks 'Add Department' button in the header;
- Application displays form to enter data;
- User leaves page;
- Data don't save to db;



The screenshot shows a web application interface. At the top, there is an orange header bar containing five navigation links: 'Main', 'Employees', 'Add Employee', 'Departments', and 'Add Department'. Below the header, the main content area has a light blue background. In the center, there is a light green rectangular box. Inside this box, the text 'Input department name:' is displayed in bold black font. Below this text is a white text input field with a thin grey border. At the bottom of the green box is a small, light grey button with the text 'ADD' in black.

Pic. 2.2 Add department

When adding an department, the following details are entered:

- Department's name;

1.3 Edit department

Main scenario:

- User clicks the 'Change' button in the departments list;
- Application displays form with existing data;
- User enters data and presses 'CHANGE' button;
- If any field are empty, error message displays;
- If all fields are field, the record is adding to db;
- If error occurs, then error message displays;

Cancel operation scenario:

- User clicks the 'Change' button in the departments list;
- Application displays form with existing data;
- User leaves page;
- Data don't save to db;



The screenshot shows a web application interface. At the top, there is an orange navigation bar with five buttons: 'Main', 'Employees', 'Add Employee', 'Departments', and 'Add Department'. Below this bar, centered on a light blue background, is a green rectangular form titled 'Input department name:'. Inside the form, there is a text input field with the placeholder text 'Retouch' and a 'CHANGE' button below it.

Pic. 2.3 Edit department

When editing a department, the following details are entered:

- Department's name;

1.4 Removing department

Main scenario:

- The user, while in the list of departments, presses the 'Delete' button in the 'ACTIONS' column;
- If the department can be removed, it's deleting from db;
- If error occurs, then error message displays;
- If department record is successfully deleted, then list of departments without deleted records is displaying;

Main Employees Add Employee Departments Add Department			
id	DEPARTMENT	AVERAGE SALARY	ACTIONS
16	Accounts and Finance	2756.5625	Delete Change
9	Admin department	2624.3814	Delete Change
2	HR	2944.6635	Delete Change
4	Infrastructures	2873.2824	Delete Change
7	IT services	2765.8415	Delete Change
6	Learning and development	2817.6061	Delete Change
8	Product development	2979.7565	Delete Change
5	Research and development	2723.7030	Delete Change
13	Retouch	2000.0000	Delete Change
3	Sales and marketing (Business development)	2914.4356	Delete Change
10	Security and transport	2813.4476	Delete Change
18	Very Strange Department	None	Delete Change

Pic. 2.4 Deleting department

Cancel operation scenario:

- Cancel operation scenario for this function doesn't exist;