МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІКТА

кафедра ЗІ



Пояснювальна записка курсового проєкту

з курсу: «Бази даних та знань»

на тему: «Розробка БД фабрики з виготовлення меблів» Варіант № 10

***Виконав:*** ст. гр. КБ-201

Коваль Р.О ***Прийняв:***доцент Гарасимчук О.І.

Львів 2023

Завдання на курсову роботу

Необхідно розробити набір програм, який дозволить створити базу даних для фабрики, що займається виробництвом меблів, доповнювати її, видаляти та редагувати записи в ній. Програма має бути робочою, відповідати усім потребам, бути зрозумілою та простою у використанні. Створити необхідні таблиці, запити, форми та звіти відповідно до індивідуального завдання.

Під час виконання необхідно:

* Проаналізувати довільну предметну галузь та провести постановку задачі на створення інформаційної системи в цій галузі;
* Створити концептуальну модель предметної галузі, логічну модель бази даних і розробити схему нормалізованої бази даних для запропонованої СУБД;
* Фізично реалізувати створену модель бази даних;
* Вибрати, обґрунтувати й реалізувати оптимальні рішення щодо розроблення інформаційних систем на основі сучасних баз даних;
* Наповнити та супроводжувати бази даних;
* Підготувати супровідну документацію.

Реферат

*Мета курсової роботи*: закріплення знань, здобутих під час вивчення дисципліни “Бази даних”, набуття навичок з проектування та розроблення баз даних, опанування програмними засобами розроблення інформаційних систем, заснованих на базах даних.

*Методи розв’язання задачі*: проведення аналізу предметної області (фабрики з виготовлення меблів), проєктування та розробка базу даних, механізмів взаємодії з нею через мови програмування PHP та SQL а також користувацького веб-інтерфейсу мовою розмітки HTML.

*Отримані результати*: розроблено програмний комплекс засобами PHP, SQL, HTML та CSS для реалізації функціювання системи управління бази даними транспортно-логістичного підприємства.

Основні відомості:

* Обсяг пояснювальної записки: 97 сторінок
* Кількість частин пояснювальної записки: 4
* Кількість ілюстрацій: 67
* Кількість джерел: 7

Ключові слова: база даних, фабрика, СУБД, база даних, веб-сайт, таблиця, схеми, форма, MYSQL, PHP, HTML, CSS.

Зміст

|  |  |
| --- | --- |
| **Завдання на курсову роботу** | **2** |
| **Реферат** | **3** |
| **Зміст** | **4** |
| **Вступ** | **5** |
| **1. Аналіз предметної області** | **6** |
| **2. Постановка задачі** | **8** |
| 1. **Проєктування бази даних**    1. **Виявлення основних сутностей предметної області** | **10** |
| **3.2. Побудова схеми бази даних** | **11** |
| 1. **Опис програмного забезпечення**    1. **Загальний опис програмного продукту** | **16** |
| **4.2. Опис архітектури програмної системи** | **17** |
| **4.3. Опис концептуальної моделі бази даних** | **18** |
| **4.4. Опис програмної реалізації** | **23** |
| **4.5. Опис задач автоматизації та інтерфейсу користувача** | **28** |
| **Висновки** | **51** |

|  |  |
| --- | --- |
| **Список використаних джерел** | **52** |
| **Додаток 1** | **53** |

**Вступ**

У даний час бази даних у тих чи інших варіаціях стали невід'ємною частиною нашого повсякденного життя та успішно використовуються практично в усіх галузях людської діяльності. У зв’язку зі стрімкими темпами зростання інформації, сучасні підприємства та організації як ніколи потребують інформаційних систем, що дозволяють упорядковувати, зберігати та ефективно використовувати наявні у них інформаційні ресурси.

Сучасний ринок меблів стрімко розвивається, що вимагає від компаній, що займаються їх виробництвом, ефективного управління та оперативної обробки великого обсягу даних. Одним з ключових інструментів, який допомагає вирішувати ці завдання, є інформаційні системи з базами даних. Фабрики, які спеціалізуються на виготовленні меблів, стикаються з низкою складних задач, пов'язаних з керуванням, плануванням, виробництвом та продажем продукції. Інформаційна система, розроблена спеціально для потреб фабрики, може значно полегшити ці процеси, забезпечуючи автоматизацію та оптимізацію багатьох рутинних завдань. Саме це й буде виконано у цій роботі.

В цій курсовій роботі для створення системи керування базами даних було використано передусім MySQL і PHP, оскільки таке поєднання технологій є досить популярним у веб-розробці та дає широкі можливості для створення баз даних.

1. Аналіз предметної області

Для створення бази даних для початку необхідно визначити, які існують проблеми зі зберіганням інформації:

* Надмірність зберігання даних. Кожен файл окремо зберігає свої дані. Інколи одна і та ж інформації міститься в кількох різних файлах чи таблицях. Приклад: і список співробітників у відділі кадрів, і список співробітників, які працюють у цеху підприємства, містять ім'я, адресу, номер телефону.
* Суперечливості. Найчастіше протиріччя є наслідком надмірності. Приклад: якщо однакові дані про одну людину представлені в двох різних записах і це "роздвоєння" не враховано, з часом записи про одну і ту ж людину відрізнятимуться.
* Не забезпечено доступ: неможливе спільне користування даними.

Використання бази даних на підприємстві є одним із найкращих рішень оптимізації даних і зберігання їх в одному місці. Також основним завданням проектування баз даних є:

* скорочення надмірності даних;
* зменшення витрат на операції оновлення, опрацювання надлишкових копій;
* усунення розбіжностей;
* спільний доступ до записів;
* цілісність даних.

Фабрика займається виготовленням меблів. Виробництво відбувається у різних цехах з певною кількістю працівників. База даних повинна містити інформацію щодо цеху і щодо працівників.

Фабрика приймає замовлення від клієнтів і повинна мати інформацію замовника.

БД повинна містити дані про виріб і матеріал, з якого виготовляють замовлення, кількість, його ціна, а також дата замовлення.

Для кожного нового клієнта необхідно виконати процес реєстрації, який включає в себе заповнення журналу клієнтів із збереженням інформації, такої як повне ім'я клієнта та його контактний номер телефону.

Клієнт оформлює замовлення, де вказано всю інформацію щодо виробу і матеріалу, також ціну, кількість і дата замовлення.

Якщо клієнт прийде наступний раз, то інформація про нього буде знайдена по його повному імені і оформиться новий договір.

Можуть з’явитися нові матеріали чи вироби, які повинні бути додані до вже існуючих. Мають бути доступні усі матеріали і вироби, розподілені по групах і для них вказані найменування, вартість у грн.

Цей програмний продукт повинен буде виконувати такі функції реєстрації клієнта, обробки замовлення клієнта, введення нових (або змінення вже існуючих) матеріалів та виробів

1. Постановка задачі

**Мета**

Реалізувати базу даних для прискорення й спрощення роботи з інформацією, яку містить фабрика:

**factory**: id, factory\_name, factory\_description, factory\_location, established\_date, number\_of\_employees, annual\_revenue

**design\_development**: id, factory\_id, employee\_name, position, project\_name,

project\_description, start\_date, end\_date, salary

**production**: id, factory\_id, employee\_name, position, section, task\_description, shift\_start, shift\_end, salary

**quality\_control**: id, factory\_id, employee\_name, position, inspection\_area, inspection\_date, issues\_found, corrective\_action, salary

**it\_department**: id, factory\_id, employee\_name, position, system\_managed, issue\_reported, resolution, date\_reported, date\_resolved, salary

**hr\_department**: id, factory\_id, employee\_name, position, hire\_date, department, performance\_review, salary

**clients**: id, username, full\_name, phone\_number, email, password **orders**: id, client\_id, product\_name, material, price, quantity, order\_date **admins**: id, username, password

Функції програми

* Додавання інформації до кожного атрибуту.
* Оновлення інформації (наприклад, оновлення ціни).
* Сортування інформації (наприклад, за ціною).
* Видалення інформації.
* Пошук інформації.
* Динамічна робота з даними і фото

Вимоги до програми

Програма має бути простою у використанні, передбачати типові помилки користувачів та мати інтуїтивний інтерфейс. Програма буде виконана у середовищі MySQL.

Концептуальна модель

Концептуальна модель предметної області полягає у структуризації наочної області: класифікуються об’єкти, фіксується сукупність атрибутів, що підлягають відображенню в БД об'єктів. Для кожного об'єкту фіксується сукупність властивостей, за допомогою яких описуватимуться конкретні екземпляри класу, і відносини (взаємозв'язки) з іншими об'єктами.

1. Проєктування бази даних
   1. **Концептуальна модель**

Для побудови бази даних використовуються такі таблиці:

1. **factory**
2. **design\_development**
3. **production**
4. **quality\_control**
5. **it\_department**
6. **hr\_department**
7. **clients**
8. **orders**
9. **admins**
   1. Побудова схеми бази даних

Концептуальна модель предметної області полягає у структуризації наочної області: класифікуються об’єкти, фіксується сукупність атрибутів, що підлягають відображенню в БД об'єктів. Для кожного об'єкту фіксується сукупність властивостей, за допомогою яких описуватимуться конкретні екземпляри класу, і відносини (взаємозв'язки) з іншими об'єктами. Потім вирішуються питання про те, яка інформація про сутності повинна бути представлена в БД і як її представити за допомогою даних.

Функціональні залежності проявляються, коли значення одного атрибута може бути визначено із значення іншого атрибута. Атрибут, який може бути визначений, називається функціонально залежним від атрибута, який є детермінантом. Коли один атрибут відношення унікально не визначає інший атрибут, але обмежує його набором зумовлених значень, це називається багатозначною залежністю.

Процес нормалізації полягає в покроковій побудові БД в нормальній формі (НФ):

* Перша нормальна форма (1НФ). Всі таблиці БД повинні задовольняти єдині вимоги — значення предметної області додатку БД не повинно мати внутрішньої структури.
* Друга нормальна форма (2НФ) — відношення знаходиться в 1НФ і кожний непервинний атрибут повністю залежить від первинного ключа.
* Третя нормальна форма (3НФ) — відношення знаходиться в 2НФ і кожен непервинний атрибут не містить транзитивних залежностей від первинного ключа.
* Нормальна форма Бойса-Кодда (**НФБК**) — відношення знаходиться в 3НФ і в ньому відсутні залежності первинних атрибутів від непервинних, тобто всі домени є можливими ключами.

1. Загальний опис програмного продукту

Для розробки було обрано Visual Studio Code — засіб для створення, редагування та опрацювання сучасних вебзастосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно, у чому його велика перевага.

Мова програмування – PHP. PHP, попередня назва: Personal Home Page Tools

* скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веброзробок. PHP підтримується переважною більшістю хостинг-провайдерів.

Мова розмітки – HTML. HTML — стандартизована мова розмітки документів для перегляду веб-сторінок у браузері. Також використано CSS. CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

Для розробки і запуску проєкта використовувалось наступне програмне забезпечення:

* + Mac OS
  + PhpStorm
  + Apache 2.4.48

- PHP 8.2.4

* + MySQL Workbench 8.0

- Google Chrome 114.0.5735.91

* + JavaScript, (SPA)
  + Bootstrap

Проєкт здатен працювати на апаратній платформі з меншими потужностями.

* 1. Опис архітектури програмної системи

Розроблена програмна система керування базою даних містить в собі наступний функціонал додавання, редагування, видалення та пошуку записів.

На UML-діаграмі нижче зображено сутності, атрибути та зв’язки між ними, описано основні функції, якими можна буде скористатися.

Для реалізації потрібного функціоналу було створено файли PHP, HTML та

CSS, JS, детальний перелік яких можна знайти у додатку 1.

* 1. Опис концептуальної моделі бази даних

Нижче, на *Рисунках 3-14*, зображено усі таблиці, створені у серидовищі MySQL та їх заповнення. Було проведено створення таблиць, налаштування за типом та довжиною даних атрибутів, а також встановлено ключі. Усі скрипти представлено у п.4.4.

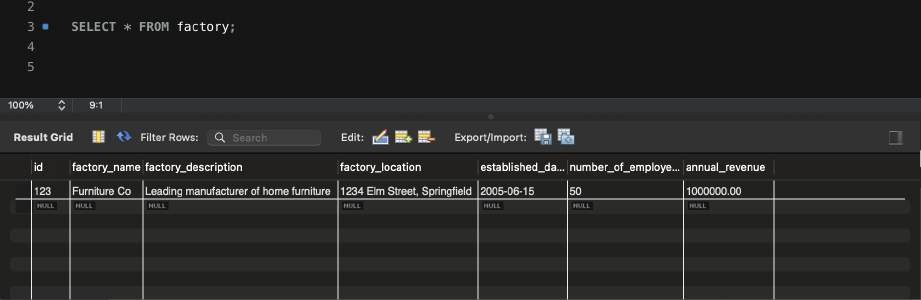


Рисунок 4.2 – Заповнена таблиця "factory"

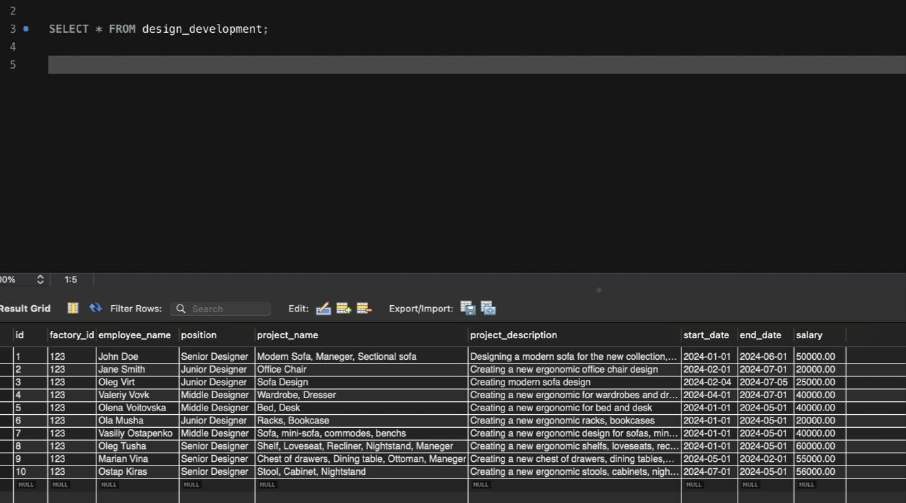


Рисунок 4.3 – Заповнена таблиця "design\_development"

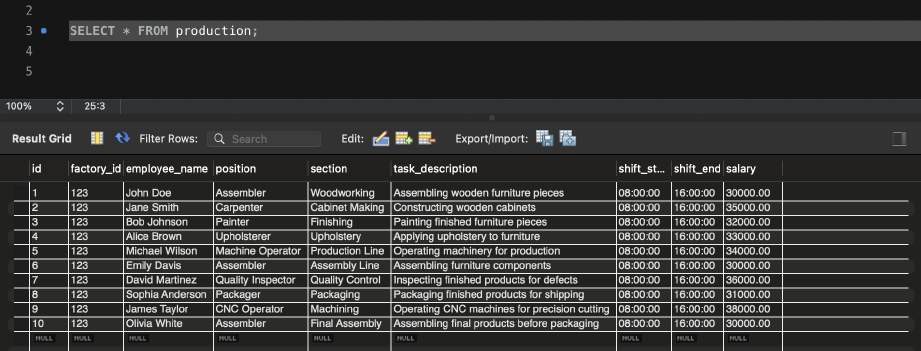


Рисунок 4.4 – Заповнена таблиця "production"

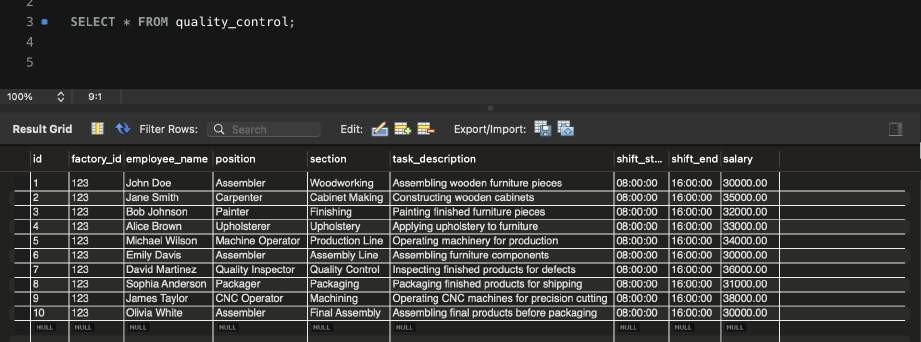


Рисунок 4.5 – Заповнена таблиця "quality\_control"

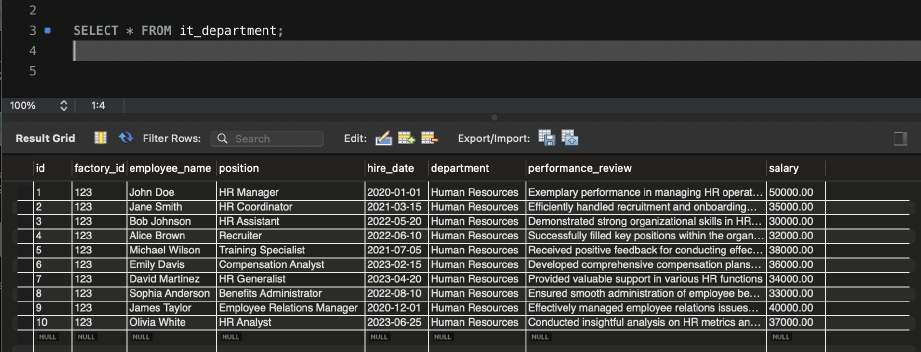


Рисунок 4.6 – Заповнена таблиця "it\_department"

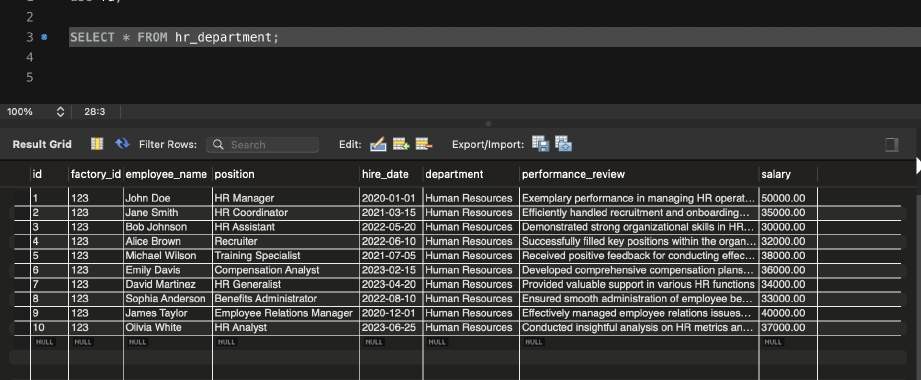


Рисунок 4.7 – Заповнена таблиця "hr\_department"

Рисунок 4.8 – Заповнена таблиця "clients"

вона заповнюється під час register, login

Рисунок 4.9 – Заповнена таблиця "orders"

вона заповнюється користувачем

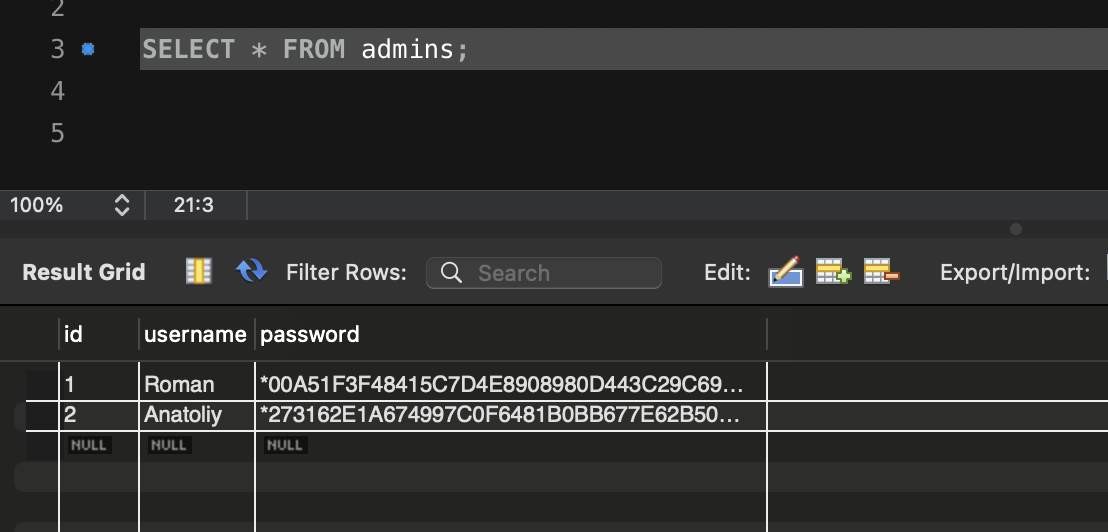


Рисунок 5 – Заповнена таблиця "admins"

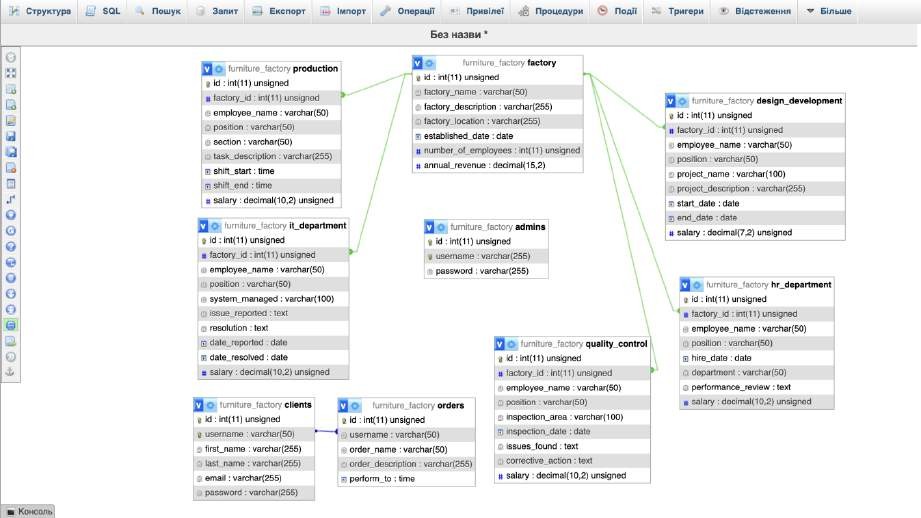


Рисунок 4.14 – Схема залежностей таблиць, створена у СУБД MySQL

* 1. Опис програмної реалізації

SQL-скрипт для створення та наповнення бази даних:

1. Скрипт для створення бази даних та таблиці «factory»: CREATE DATABASE IF NOT EXISTS furniture\_factory; USE furniture\_factory;

CREATE TABLE IF NOT EXISTS factory ( id INT(11) UNSIGNED NOT NULL,

factory\_name VARCHAR(50) NOT NULL, factory\_description VARCHAR(255) NOT NULL, factory\_location VARCHAR(255) NOT NULL, established\_date DATE NOT NULL, number\_of\_employees INT(11) UNSIGNED NOT NULL, annual\_revenue DECIMAL(15,2) NOT NULL,

PRIMARY KEY(id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «design\_development»:

CREATE TABLE IF NOT EXISTS design\_development (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

factory\_id INT(11) UNSIGNED NOT NULL, employee\_name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL,

project\_name VARCHAR(100) NOT NULL, project\_description VARCHAR(255), start\_date DATE NOT NULL,

end\_date DATE,

salary DECIMAL(10, 2) UNSIGNED NOT NULL,

INDEX fk\_factory\_id (factory\_id),

FOREIGN KEY (factory\_id) REFERENCES factory(id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «production»:

CREATE TABLE IF NOT EXISTS production (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

factory\_id INT(11) UNSIGNED NOT NULL, employee\_name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL, section VARCHAR(50) NOT NULL,

task\_description VARCHAR(255), shift\_start TIME NOT NULL, shift\_end TIME NOT NULL,

salary DECIMAL(10, 2) UNSIGNED NOT NULL,

INDEX fk\_factory\_id (factory\_id),

FOREIGN KEY (factory\_id) REFERENCES factory(id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «quality\_control»:

CREATE TABLE IF NOT EXISTS quality\_control (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

factory\_id INT(11) UNSIGNED NOT NULL, employee\_name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL,

inspection\_area VARCHAR(100) NOT NULL, inspection\_date DATE NOT NULL, issues\_found TEXT,

corrective\_action TEXT,

salary DECIMAL(10, 2) UNSIGNED NOT NULL,

INDEX fk\_factory\_id (factory\_id),

FOREIGN KEY (factory\_id) REFERENCES factory(id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «it\_department»:

CREATE TABLE IF NOT EXISTS it\_department (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

factory\_id INT(11) UNSIGNED NOT NULL, employee\_name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL,

system\_managed VARCHAR(100) NOT NULL, issue\_reported TEXT,

resolution TEXT, date\_reported DATE NOT NULL,

date\_resolved DATE,

salary DECIMAL(10, 2) UNSIGNED NOT NULL,

INDEX fk\_factory\_id (factory\_id),

FOREIGN KEY (factory\_id) REFERENCES factory(id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «hr\_department»:

CREATE TABLE IF NOT EXISTS hr\_department (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

factory\_id INT(11) UNSIGNED NOT NULL, employee\_name VARCHAR(50) NOT NULL, position VARCHAR(50) NOT NULL,

hire\_date DATE NOT NULL, department VARCHAR(50) NOT NULL, performance\_review TEXT,

salary DECIMAL(10, 2) UNSIGNED NOT NULL,

UNIQUE(unique\_key),

INDEX fk\_factory\_id (factory\_id),

FOREIGN KEY (factory\_id) REFERENCES factory(id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «clients»:

CREATE TABLE IF NOT EXISTS clients (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(50) NOT NULL,

first\_name VARCHAR(255) NOT NULL, last\_name VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL,

UNIQUE(username)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «orders»:

CREATE TABLE IF NOT EXISTS orders (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(50) NOT NULL,

order\_name VARCHAR(50) NOT NULL, order\_description VARCHAR(255) NOT NULL, perform\_to TIME NOT NULL,

INDEX fk\_username (username),

FOREIGN KEY (username) REFERENCES clients(username)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

1. Скрипт для створення таблиці «admins»:

CREATE TABLE IF NOT EXISTS admins (

id INT(11) UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL,

UNIQUE(username)

)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

* 1. Наповнення бази даних

Скрипт для заповнення таблиць бази даних:

* 1. Опис задач автоматизації та інтерфейсу користувача

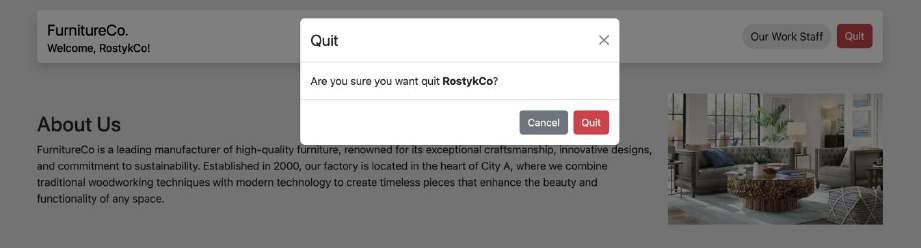
Користувацький інтерфейс – це створений вебсайт, що використовує мови HTML та PHP та розроблений у середовищі Microsoft VS Code з використанням бази даних попередньо створеної у СУБД MySQL. Сайт є швидкодійним та зручним у використанні. Він доступний для використання у будьякому сучасному браузері

Основні функції веб-сайту:

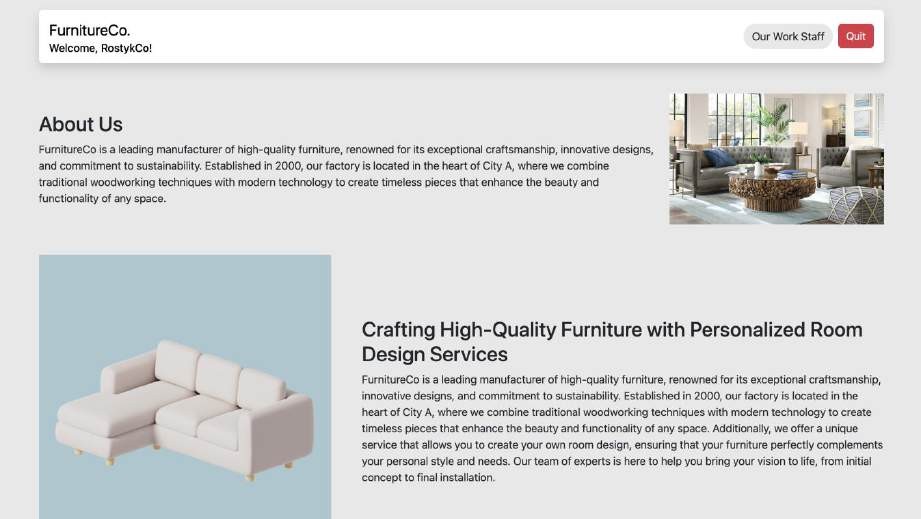
* Виведення інформації про таблиці та їх вміст;
* Додавання, видалення, та оновлення та інформації;
* Сортування інформаційних ресурсів;

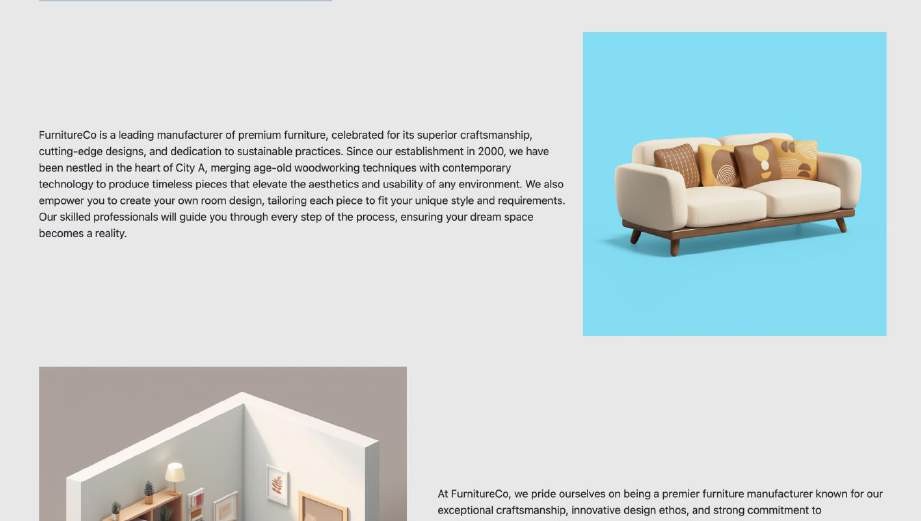
Сайт має інтуїтивно зрозумілий інтерфейс, тому для його використання не потрібні спеціальні навички від користувача. Приємний зовнішній вигляд, що був налаштований за допомогою мови стилів CSS додає зручності та приємного досвіду використання.

Після запуску програми користувача зустрічає домашня сторінка, на якій можна переглянути всі збережені дані а також переміститись до потрібної таблиці за допомогою кнопок. На рисунках 16-21 зображено початковий екран та усі 6 таблиць, доступні за натисканням. При наведенні кнопки підсвічуються, що візуально простіше сприймається користувачем.









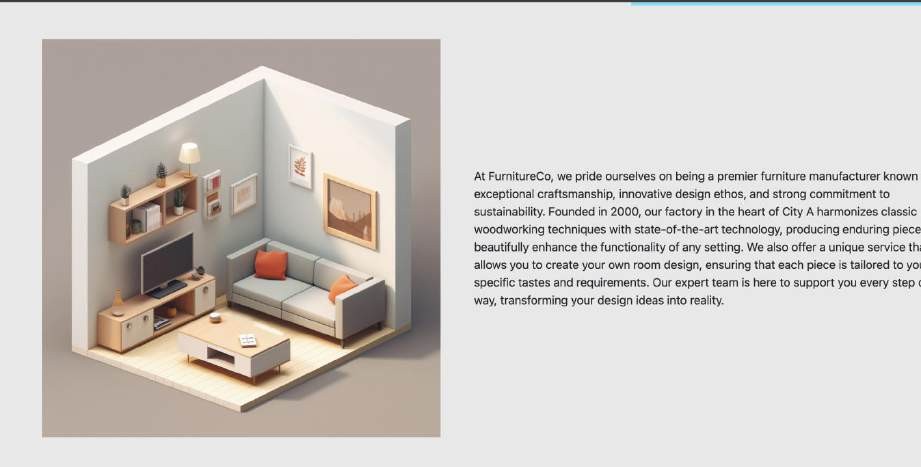
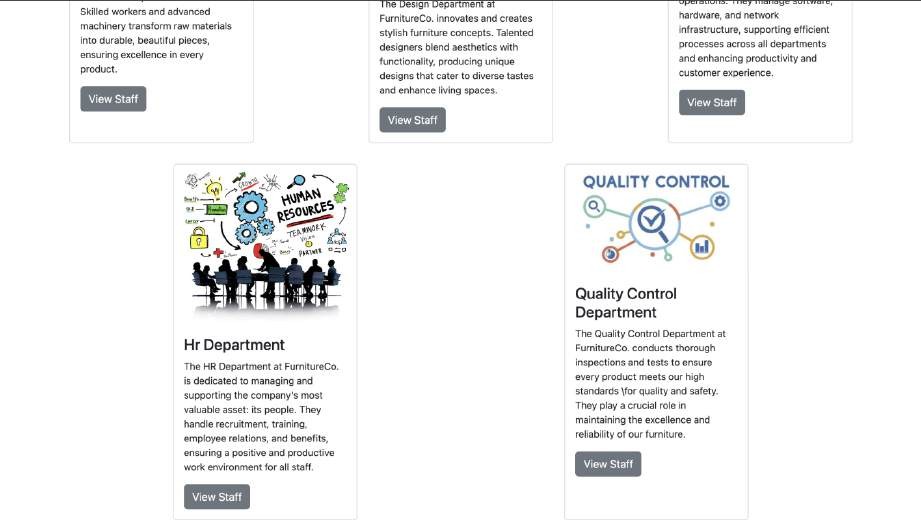
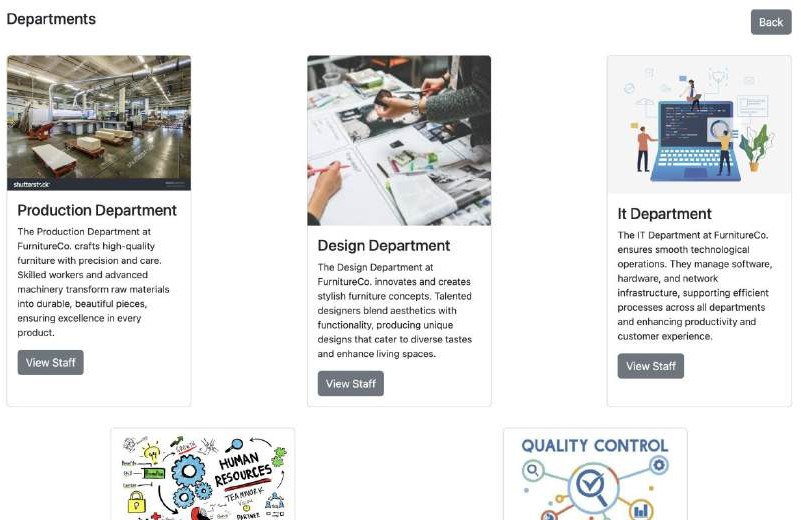
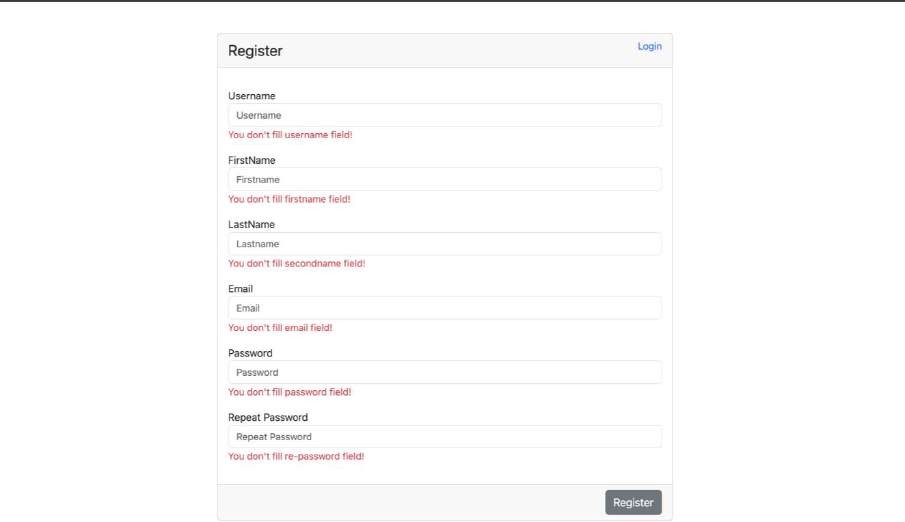


Рисунок 4.15 – Головне меню (колонка Number Of Employee відображає

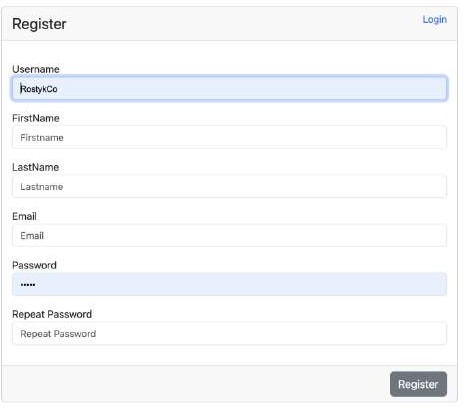


реальну кількість робітників)

Рисунок 4.16 – Меню відділень







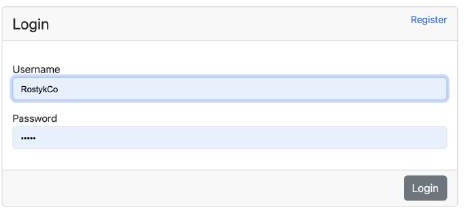
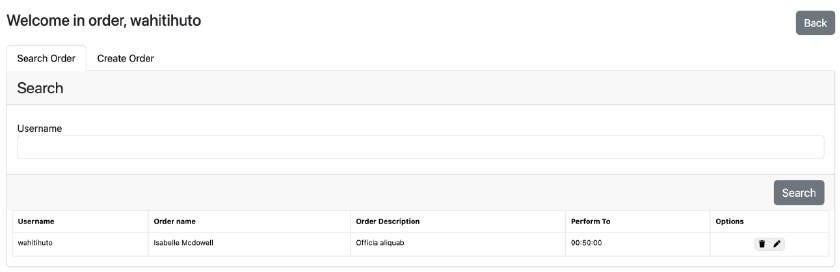


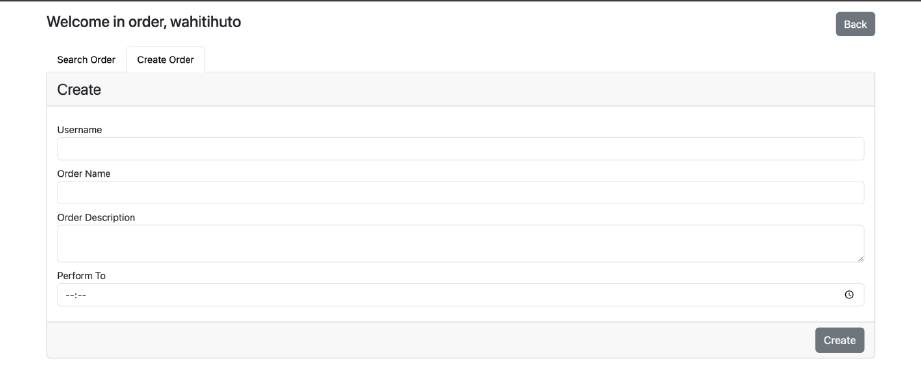
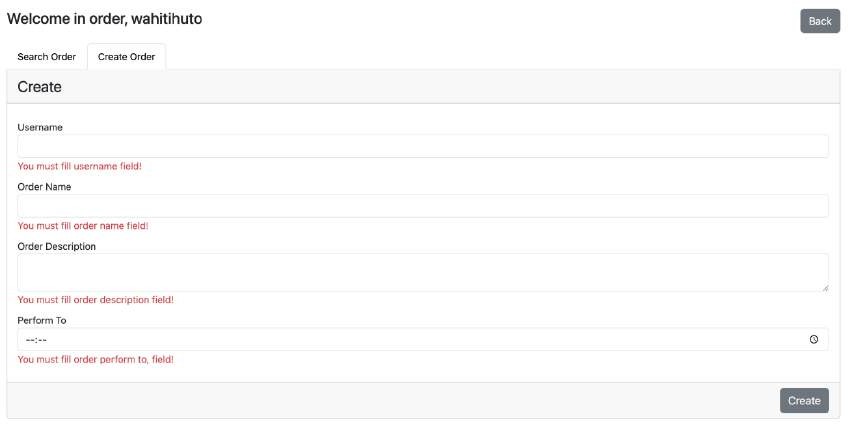
Рисунок 4.17 – Авторизація (Логіка в тому, що

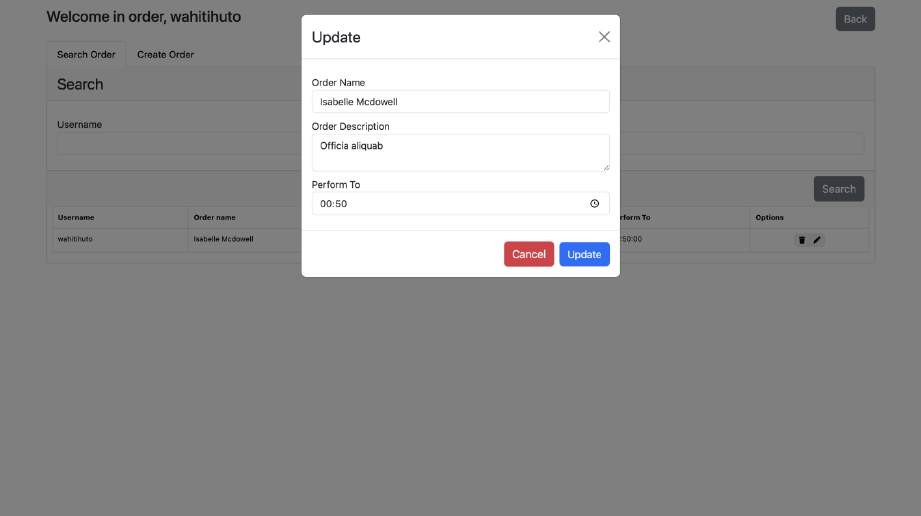
admin може робити зміни з відділами, а також немає доступу до замовлень)











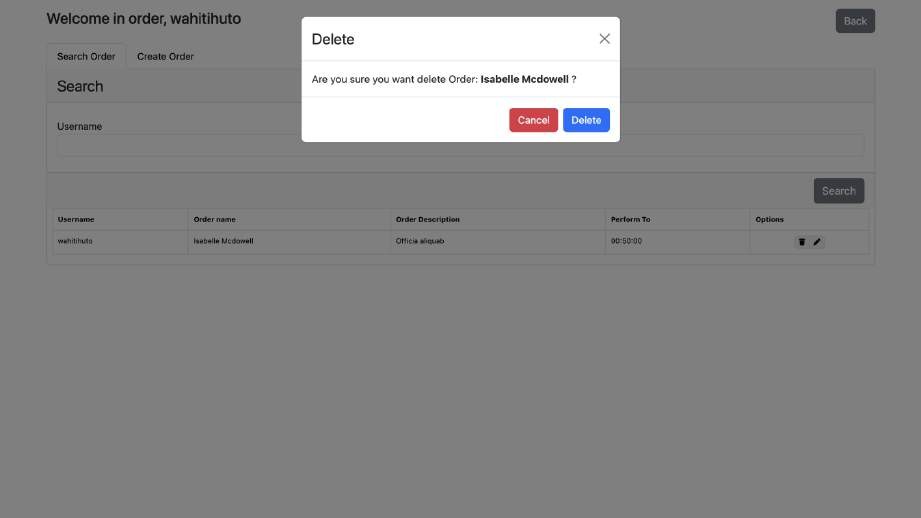
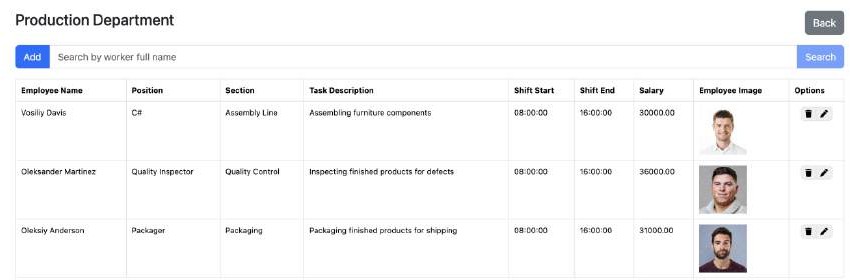
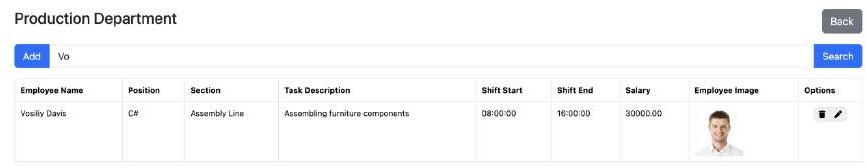
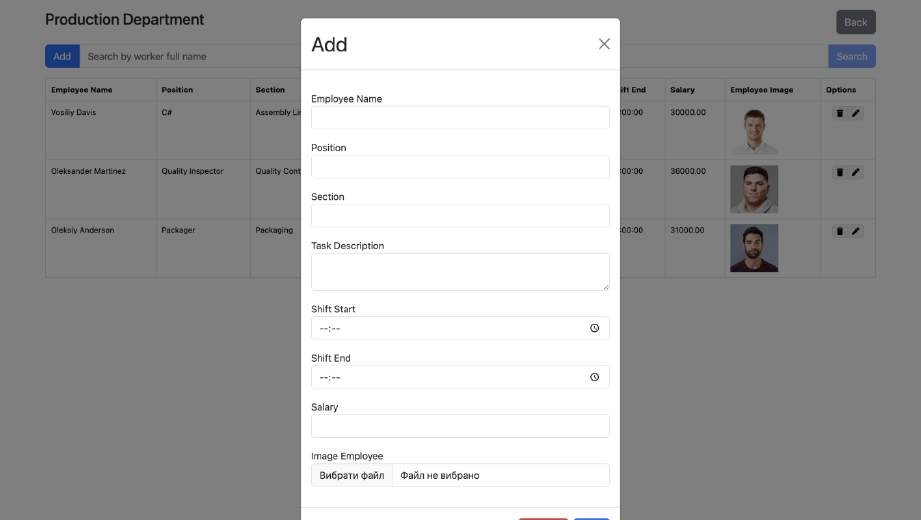


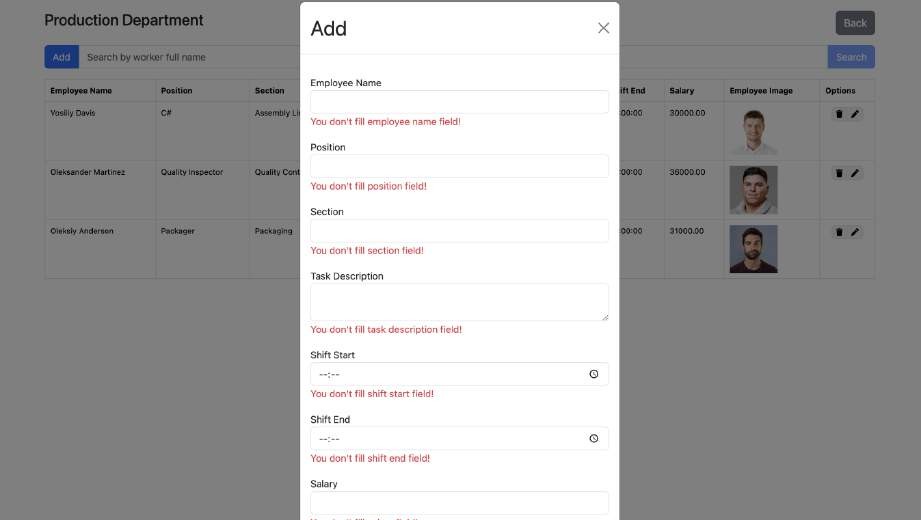
Рисунок 4.18 – Замовлення (Доступно лише для client і то, тільки для того, який авторизований)

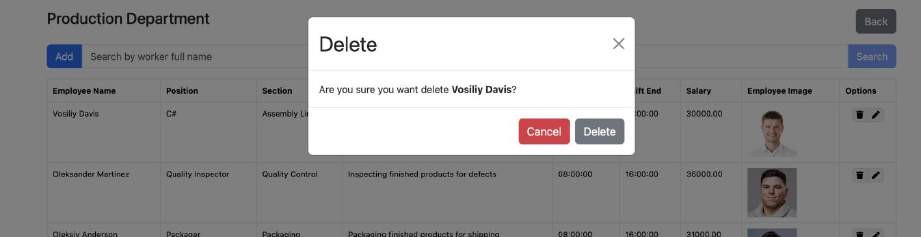












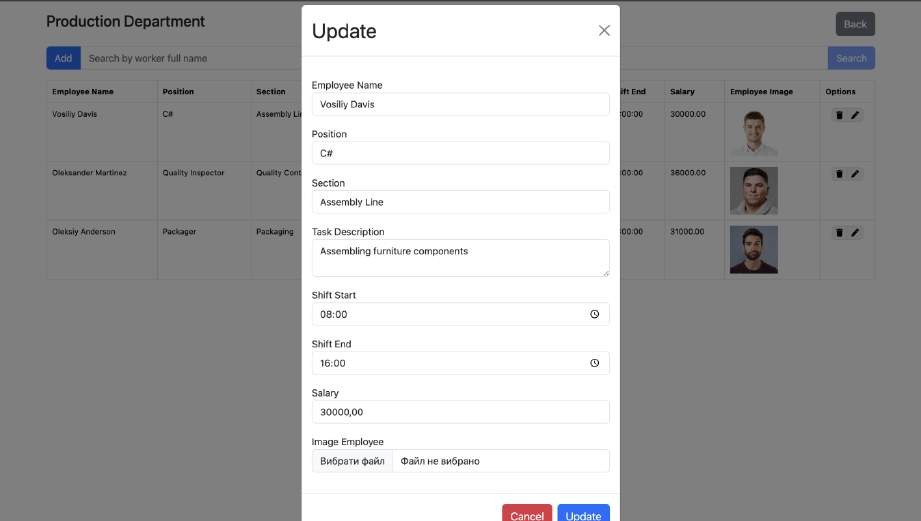
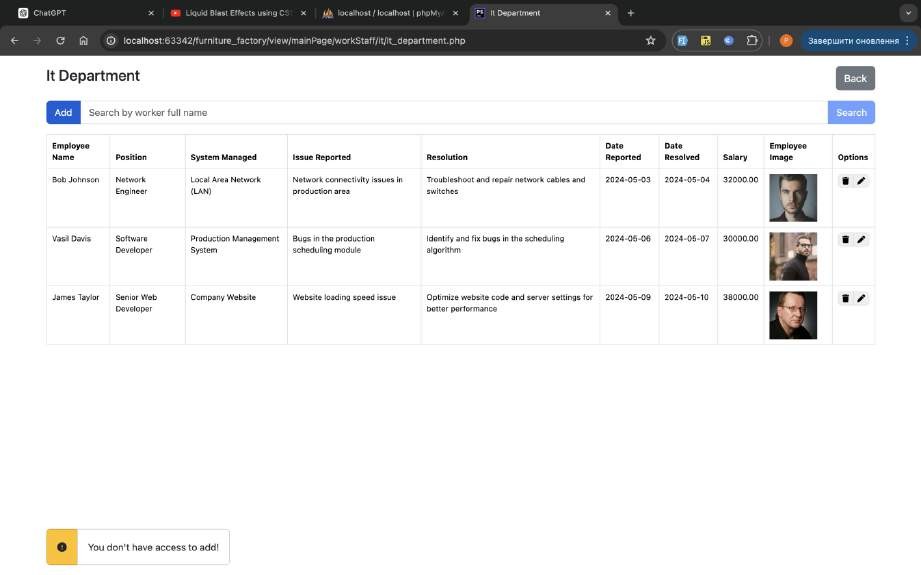


Рисунок 4.19 – Дії з відділеннями (Доступно лише для admin, також показ таблиці, пошук)



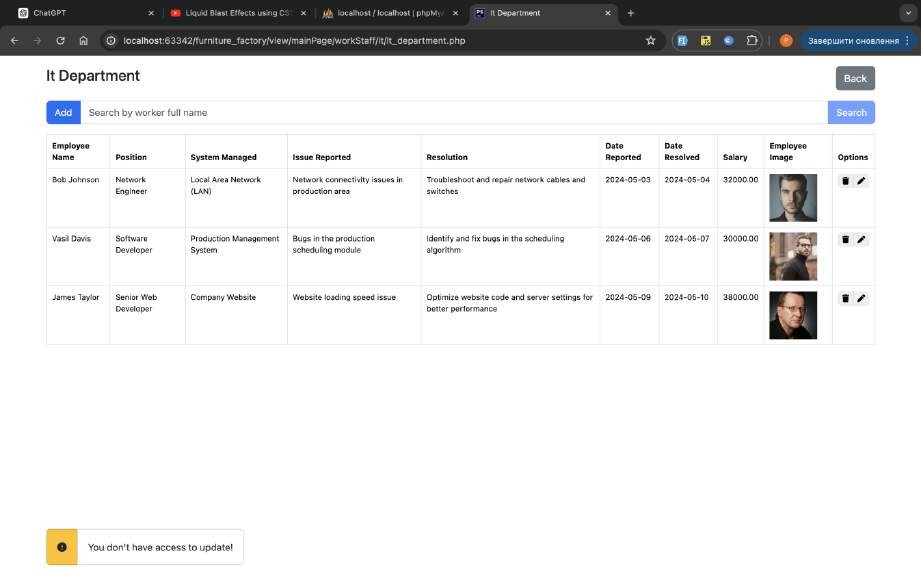
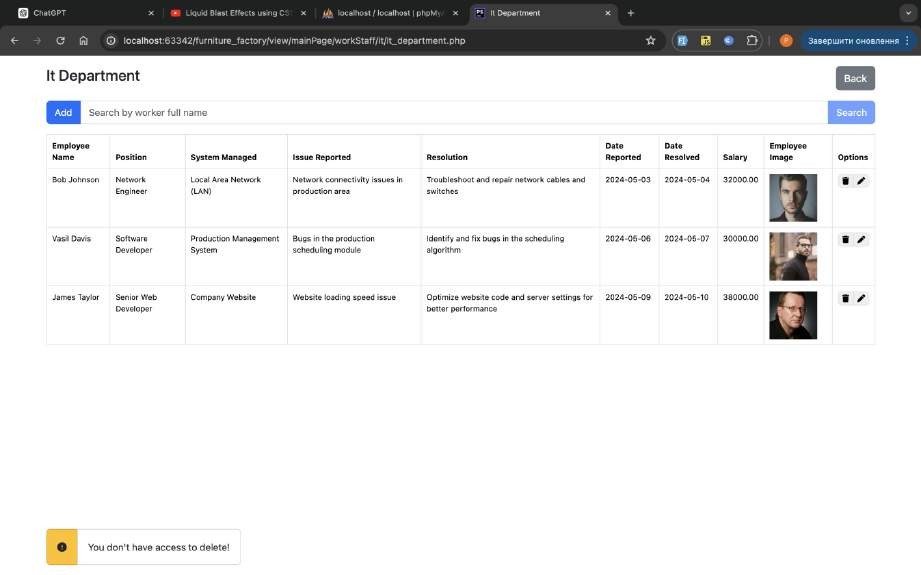


Рисунок 4.20 – Попередження для client якшо він намагається зробити дію

Висновки

В межах цієї курсової роботи мною була створена база даних в середовищі MySQL для організації даних фабрики з виготовлення меблів. Було розроблено програмне забезпечення у вигляді веб-сайту, створеного на мовах PHP та HTML. В процесі виконання цієї роботи мною було проведено аналіз предметної області, під час нього з’ясовано основні функції фабрики з виготовлення меблів і проблеми які можуть виникнути для такої фабрики. Також була побудована і спроєктована схема бази даних, після чого проведена нормалізація та приведення таблиць до 3-ої нормальної форми і нормальної форми Бойса-Кодда.

Результатом даної роботи є реляційна схема бази даних та концептуальна модель даних. Крім того, були розроблені специфікації програмних модулів, включаючи їх алгоритми та графічний інтерфейс.

При роботі з програмою, що була розроблена, користувачеві надається зручна навігація по базі даних. Наявні таблиці "Замовлення ", "Клієнт ", "Виріб", "Матеріал", "Цех" та "Працівники". До кожної таблиці реалізовані операції додавання, видалення, пошуку, оновлення та сортування даних, до кожної створено форми, до яких є доступ по кнопці, за допомогою яких можна взаємодіяти з даними. Хороше візуальне оформлення, створене за допомогою CSS робить використання інтуїтивним та легким.

Список використаних джерел

1. Стаття даних про HTML https://uk.wikipedia.org/wiki/HTML
2. Стаття даних про CSS https://uk.wikipedia.org/wiki/CSS
3. Стаття даних про PHP https://uk.wikipedia.org/wiki/PHP
4. Лекційні матеріали з курсу «бази даних та знань» Національного університету

«Львівська політехніка»

https://vns.lpnu.ua/course/view.php?id=6130

1. Документація по PHP https://[www.php.net/manual/en/](http://www.php.net/manual/en/)
2. Документація по HTML https://developer.mozilla.org/en-US/docs/Web/HTML
3. Документація по CSS <https://developer.mozilla.org/en-US/docs/Web/CSS>

Додаток 1

**sqlMain.php**

<?php

class MainSql{

private $connection = null; private $username = 'root'; private $server = '127.0.0.1'; private $password = '';

private $db\_name = 'furniture\_factory';

public function construct()

{

$this->connection = new mysqli($this->server, $this->username, $this->password, $this->db\_name);

$this->selectDatabase();

}

private function selectDatabase()

{

mysqli\_select\_db($this->connection, $this->db\_name);

}

// register

public function checkUsernameClients($username){

$checkUsername = $this->connection->prepare('SELECT \* FROM clients where username = ?');

$checkUsername->bind\_param('s', $username);

if($checkUsername->execute() !== TRUE){ http\_response\_code(500);

die('Error while check username!');

}

else{

$result = $checkUsername->get\_result();

$num\_row = $result->num\_rows;

$checkUsername->close();

return $num\_row > 0;

}

}

public function checkUsernameAdmin($username){

$checkAdmin = $this->connection->prepare('SELECT \* FROM admins where username = ?');

$checkAdmin->bind\_param('s', $username);

if($checkAdmin->execute() !== TRUE){ http\_response\_code(500);

die('Error while check username!');

}

else{

$result = $checkAdmin->get\_result();

$num\_row = $result->num\_rows;

$checkAdmin->close();

return $num\_row > 0;

}

}

public function addClient($request){

$username = $request['username'];

$firstName = $request['firstname'];

$lastname = $request['lastname'];

$email = $request['email'];

$password = $request['password'];

$addNewClient = $this->connection->prepare('INSERT INTO clients (`username`, `first\_name`,

`last\_name`, `email`, `password`) VALUES (?, ?, ?, ?, PASSWORD(?))');

$addNewClient->bind\_param('sssss', $username, $firstName, $lastname, $email, $password);

if($addNewClient->execute() !== TRUE){ http\_response\_code(500);

die('Error while add client!');

}

else{

$addNewClient->close(); return true;

}

}

// login check password

public function checkPasswordAdmin($username, $password){

$checkAdmin = $this->connection->prepare('SELECT \* FROM admins where username = ? and password = PASSWORD(?)');

$checkAdmin->bind\_param('ss', $username, $password);

if($checkAdmin->execute() !== TRUE){ http\_response\_code(500);

die('Error while check username and password!');

}

else{

$result = $checkAdmin->get\_result();

$num\_row = $result->num\_rows;

$checkAdmin->close();

return $num\_row === 1;

}

}

public function checkPasswordClient($username, $password){

$checkClient = $this->connection->prepare('SELECT \* FROM clients where username = ? and password = PASSWORD(?)');

$checkClient->bind\_param('ss', $username, $password);

if($checkClient->execute() !== TRUE){ http\_response\_code(500);

die('Error while check username and password!');

}

else{

$result = $checkClient->get\_result();

$num\_row = $result->num\_rows;

$checkClient->close();

session\_start();

$\_SESSION['username'] = $username; session\_write\_close();

return $num\_row === 1;

}

}

public function getUsername()

{

session\_start();

$user = $\_SESSION['username']; session\_write\_close();

return $user;

}

// get data

public function getData($table){

$getData = $this->connection->prepare("SELECT \* FROM $table");

$res\_array = [];

if($getData->execute() !== TRUE){ http\_response\_code(500);

die('Error while select data from quality control table!');

}

else{

$result = $getData->get\_result();

if($result->num\_rows > 0){

while($row = $result->fetch\_assoc()){

$res\_array[] = $row;

}

}

$getData->close();

return $res\_array;

}

}

// orders, orders actions

// check username in client i use current checkUsernameClients public function getOrdersFor($username){

$getData = $this->connection->prepare('SELECT \* FROM orders where username = ?');

$getData->bind\_param('s', $username);

$res\_array = [];

if($getData->execute() !== TRUE){ http\_response\_code(500);

die('Error while select data from order for current user!');

}

else{

$result = $getData->get\_result(); if($result->num\_rows > 0){

while($row = $result->fetch\_assoc()){

$res\_array[] = $row;

}

}

else{

return ['empty\_message' => "This user don't have orders!"];

}

$getData->close(); return $res\_array;

}

}

public function addOrder($request){

$username = $request['create\_username'];

$order\_name = $request['create\_order\_name'];

$order\_description = htmlspecialchars($request['create\_order\_description']);

$perform\_to = $request['create\_perform\_to'];

$add\_order = $this->connection->prepare('INSERT INTO `orders` (`username`, `order\_name`, `order\_description`, `perform\_to`) VALUES (?, ?, ?, ?)');

$add\_order->bind\_param('ssss', $username, $order\_name, $order\_description, $perform\_to);

if($add\_order->execute() !== TRUE){ http\_response\_code(500);

die('Error while add to order!');

}

$add\_order->close();

return true;

}

public function updateOrder($request)

{

$id = (int)$request['id'];

$order\_name = $request['update\_order\_name'];

$order\_description = htmlspecialchars($request['update\_order\_description']);

$perform\_to = $request['update\_perform\_to'];

$update\_order = $this->connection->prepare('UPDATE orders SET order\_name = ?, order\_description = ?,

perform\_to = ? where id = ?');

$update\_order->bind\_param('sssi', $order\_name, $order\_description, $perform\_to, $id);

if($update\_order->execute() !== TRUE){

http\_response\_code(500); die('Error while update orders!');

}

else{

$update\_order->close();

$selectUpdatedOrder = $this->connection->prepare('SELECT \* FROM orders where id = ?');

$selectUpdatedOrder->bind\_param('i', $id);

if($selectUpdatedOrder->execute() !== TRUE){ http\_response\_code(500);

die('Error while select current order!');

}

else{

$result = $selectUpdatedOrder->get\_result();

if($result->num\_rows > 0){

$row = $result->fetch\_assoc();

$selectUpdatedOrder->close(); return $row;

}

}

}

}

function deleteOrder($id){

$id = (int)$id;

$deleteOrder = $this->connection->prepare('DELETE FROM orders where id = ?');

$deleteOrder->bind\_param('i', $id);

if($deleteOrder->execute() !== TRUE){

http\_response\_code(500);

die('Error while delete order form orders!');

}

else{

$deleteOrder->close(); return true;

}

}

// actions on tables

public function selectFactoryId(){

$getFactoryIdent = $this->connection->prepare('SELECT id FROM factory;');

if($getFactoryIdent->execute() !== TRUE){

http\_response\_code(500); die('Error while select factory id');

}

else{

$result = $getFactoryIdent->get\_result(); if($result->num\_rows > 0){

$data = $result->fetch\_assoc();

$getFactoryIdent->close(); return (int)$data['id'];

}

}

}

public function getRow($table, $id){

$id = (int)$id;

$get = $this->connection->prepare("SELECT \* FROM $table where id = ?");

$get->bind\_param('i', $id);

if($get->execute() !== TRUE){ http\_response\_code(500); die('Error while after action');

}

else{

$result = $get->get\_result();

if($result->num\_rows > 0){

$row = $result->fetch\_assoc();

$get->close(); return $row;

}

else{

return null;

}

}

}

// search

public function search($value, $table)

{

$value = '%'. $value . '%';

$search = $this->connection->prepare("SELECT id FROM $table where employee\_name LIKE ?");

$search->bind\_param('s', $value);

$res\_array = [];

if($search->execute() !== TRUE){ http\_response\_code(500);

die('Error while search!');

}

else{

$result = $search->get\_result(); if($result->num\_rows > 0){

while ($row = $result->fetch\_assoc()){

$res\_array[] = $row;

}

}

$search->close();

return $res\_array;

}

}

public function delete($id, $table){

$id = (int)$id;

$delete = $this->connection->prepare("DELETE FROM $table where id = ?");

$delete->bind\_param('i', $id);

if($delete->execute() !== TRUE){ http\_response\_code(500); die('Error while delete form table!');

}

else{

$delete->close();

return true;

}

}

// design department

public function addDesign($request, $table){

$factory\_id = $this->selectFactoryId();

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$project\_name = $request['project\_name'];

$project\_description = htmlspecialchars($request['project\_description']);

$start\_date = $request['start\_date'];

$end\_date = $request['end\_date'];

$salary = (int)$request['salary'];

$add = $this->connection->prepare("INSERT INTO $table (`factory\_id`,

`employee\_name`, `position`, `project\_name`, `project\_description`,

`start\_date`, `end\_date`, `salary`) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

$add->bind\_param('issssssi', $factory\_id,

$employee\_name, $position, $project\_name, $project\_description, $start\_date,

$end\_date, $salary);

if($add->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$row\_id = $add->insert\_id;

$add->close();

return $this->getRow($table, $row\_id);

}

}

public function updateDesign($request, $table){

$id\_update = (int)$request['update\_id'];

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$project\_name = $request['project\_name'];

$project\_description = htmlspecialchars($request['project\_description']);

$start\_date = $request['start\_date'];

$end\_date = $request['end\_date'];

$salary = (int)$request['salary'];

$update = $this->connection->prepare("UPDATE $table SET employee\_name = ? , position = ?, project\_name = ?, project\_description = ?, start\_date = ?, end\_date = ?, salary = ? where id = ?");

$update->bind\_param('ssssssii',

$employee\_name, $position, $project\_name, $project\_description, $start\_date,

$end\_date, $salary, $id\_update);

if($update->execute() !== TRUE){ http\_response\_code(500); die('Error while update!');

}

else{

$update->close();

return $this->getRow($table, $id\_update);

}

}

// hr department

function addHr($request, $table)

{

$factory\_id = $this->selectFactoryId();

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$hire\_date = $request['hire\_date'];

$department = $request['department'];

$performance\_review = htmlspecialchars($request['performance\_review']);

$salary = (int)$request['salary'];

$add = $this->connection->prepare("INSERT INTO $table (`factory\_id`,

`employee\_name`, `position`, `hire\_date`, `department`,

`performance\_review`, `salary`) VALUES (?, ?, ?, ?, ?, ?, ?)");

$add->bind\_param('isssssi', $factory\_id,

$employee\_name, $position, $hire\_date, $department, $performance\_review, $salary);

if($add->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$row\_id = $add->insert\_id;

$add->close();

return $this->getRow($table, $row\_id);

}

}

function updateHr($request, $table)

{

$id\_update = (int)$request['update\_id'];

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$hire\_date = $request['hire\_date'];

$department = $request['department'];

$performance\_review = htmlspecialchars($request['performance\_review']);

$salary = (int)$request['salary'];

$update = $this->connection->prepare("UPDATE $table SET employee\_name = ?, position = ?, hire\_date = ?, department = ?, performance\_review = ?, salary = ? where id = ?");

$update->bind\_param('sssssii',

$employee\_name, $position, $hire\_date, $department, $performance\_review, $salary, $id\_update);

if($update->execute() !== TRUE){ http\_response\_code(500); die('Error while update!');

}

else{

$update->close();

return $this->getRow($table, $id\_update);

}

}

// it department

function addIt($request, $table)

{

$factory\_id = $this->selectFactoryId();

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$system\_managed = $request['system\_managed'];

$issue\_reported = htmlspecialchars($request['issue\_reported']);

$resolution = htmlspecialchars($request['resolution']);

$date\_reported = $request['date\_reported'];

$date\_resolved = $request['date\_resolved'];

$salary = (int)$request['salary'];

$add = $this->connection->prepare("INSERT INTO $table (`factory\_id`,

`employee\_name`, `position`, `system\_managed`, `issue\_reported`,

`resolution`, `date\_reported`, `date\_resolved`, `salary`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");

$add->bind\_param('isssssssi', $factory\_id,

$employee\_name, $position, $system\_managed, $issue\_reported,

$resolution, $date\_reported, $date\_resolved, $salary);

if($add->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$row\_id = $add->insert\_id;

$add->close();

return $this->getRow($table, $row\_id);

}

}

function updateIt($request, $table)

{

$id\_update = (int)$request['update\_id'];

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$system\_managed = $request['system\_managed'];

$issue\_reported = htmlspecialchars($request['issue\_reported']);

$resolution = htmlspecialchars($request['resolution']);

$date\_reported = $request['date\_reported'];

$date\_resolved = $request['date\_resolved'];

$salary = (int)$request['salary'];

$update = $this->connection->prepare("UPDATE $table SET employee\_name = ?, position = ?, system\_managed = ?, issue\_reported = ?, resolution = ?, date\_reported = ?, date\_resolved = ?, salary = ? where id = ?");

$update->bind\_param('sssssssii',

$employee\_name, $position, $system\_managed, $issue\_reported,

$resolution, $date\_reported, $date\_resolved, $salary, $id\_update);

if($update->execute() !== TRUE){ http\_response\_code(500); die('Error while update!');

}

else{

$update->close();

return $this->getRow($table, $id\_update);

}

}

// production

function addProduction($request, $table)

{

$factory\_id = $this->selectFactoryId();

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$section = $request['section'];

$task\_description = htmlspecialchars($request['task\_description']);

$shift\_start = $request['shift\_start'];

$shift\_end = $request['shift\_end'];

$salary = (int)$request['salary'];

$add = $this->connection->prepare("INSERT INTO $table (`factory\_id`,

`employee\_name`, `position`, `section`, `task\_description`,

`shift\_start`, `shift\_end`, `salary`) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

$add->bind\_param('issssssi', $factory\_id,

$employee\_name, $position, $section, $task\_description,

$shift\_start, $shift\_end, $salary);

if($add->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$row\_id = $add->insert\_id;

$add->close();

return $this->getRow($table, $row\_id);

}

}

function updateProduction($request, $table)

{

$id\_update = (int)$request['update\_id'];

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$section = $request['section'];

$task\_description = htmlspecialchars($request['task\_description']);

$shift\_start = $request['shift\_start'];

$shift\_end = $request['shift\_end'];

$salary = (int)$request['salary'];

$update = $this->connection->prepare("UPDATE $table SET employee\_name = ?, position = ?, section = ?, task\_description = ?, shift\_start = ?, shift\_end = ?, salary = ? where id = ?");

$update->bind\_param('ssssssii',

$employee\_name, $position, $section, $task\_description,

$shift\_start, $shift\_end, $salary, $id\_update);

if($update->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$update->close();

return $this->getRow($table, $id\_update);

}

}

// quality control

function addQualityControl($request, $table)

{

$factory\_id = $this->selectFactoryId();

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$inspection\_area = $request['inspection\_area'];

$inspection\_date = $request['inspection\_date'];

$issues\_found = htmlspecialchars($request['issues\_found']);

$corrective\_action = htmlspecialchars($request['corrective\_action']);

$salary = (int)$request['salary'];

$add = $this->connection->prepare("INSERT INTO $table (`factory\_id`,

`employee\_name`, `position`, `inspection\_area`, `inspection\_date`,

`issues\_found`, `corrective\_action`, `salary`) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

$add->bind\_param('issssssi', $factory\_id,

$employee\_name, $position, $inspection\_area, $inspection\_date,

$issues\_found, $corrective\_action, $salary);

if($add->execute() !== TRUE){ http\_response\_code(500); die('Error while add!');

}

else{

$row\_id = $add->insert\_id;

$add->close();

return $this->getRow($table, $row\_id);

}

}

function updateQualityControl($request, $table)

{

$id\_update = (int)$request['update\_id'];

$employee\_name = $request['employee\_name'];

$position = $request['position'];

$inspection\_area = $request['inspection\_area'];

$inspection\_date = $request['inspection\_date'];

$issues\_found = $request['issues\_found'];

$corrective\_action = htmlspecialchars($request['corrective\_action']);

$salary = (int)$request['salary'];

$update = $this->connection->prepare("UPDATE $table SET employee\_name = ?, position = ?, inspection\_area = ?, inspection\_date = ?, issues\_found = ?, corrective\_action = ?, salary = ? where id = ?");

$update->bind\_param('ssssssii',

$employee\_name, $position, $inspection\_area, $inspection\_date,

$issues\_found, $corrective\_action, $salary, $id\_update);

if($update->execute() !== TRUE){ http\_response\_code(500); die('Error while update!');

}

else{

$update->close();

return $this->getRow($table, $id\_update);

}

}

// sum workers

public function getSumWorkers() {

$sql = '

SELECT COUNT(\*) as count FROM it\_department UNION ALL

SELECT COUNT(\*) as count FROM hr\_department UNION ALL

SELECT COUNT(\*) as count FROM production UNION ALL

SELECT COUNT(\*) as count FROM quality\_control UNION ALL

SELECT COUNT(\*) as count FROM design\_development';

$statement = $this->connection->prepare($sql); if($statement->execute() !== TRUE){

http\_response\_code(500); die('Error while get sum workers!');

}

$totalCount = 0;

$results = $statement->get\_result(); if($results->num\_rows > 0){

while($row = $results->fetch\_assoc()){

$totalCount += (int)$row['count'];

}

}

return $totalCount;

}

// check access to actions

public function checkAccessToAction($username)

{

$checkAccess = $this->connection->prepare('SELECT \* FROM admins where username = ?');

$checkAccess->bind\_param('s', $username);

if($checkAccess->execute() !== TRUE){ http\_response\_code(500);

die('Error while check access!');

}

else{

$result = $checkAccess->get\_result();

if($result->num\_rows > 0){ return true;

}

return false;

}

}

}

Login.php

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../bootstrap/style.php');?>

<link rel="stylesheet" href="./style.css">

<title>Login</title>

</head>

<body>

<div class="row mx-2">

<div id="card\_login" class="card col-md-6 offset-md-3 px-0 mt-5" >

<form id="form\_login">

<div class="card-header d-flex justify-content-between">

<div class="modal-title text-title">Login</div>

<a href="http://localhost:63342/furniture\_factory/auth/register/register.php" class="text text-

decoration-none text-custom" >Register</a>

</div>

<div class="card-body">

<div class="my-3" >

<label for="username">Username</label>

<input autocomplete="false" type="text" name="username" id="username" placeholder="Username" class="form-control text-custom" autofocus>

<span id="err\_username\_login" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="password">Password</label>

<input autocomplete="false" type="password" name="password" id="password" placeholder="Password" class="form-control text-custom">

<span id="err\_password\_login" class="text-custom text text-danger"></span>

</div>

</div>

<div class="card-footer text-end">

<button class="btn btn-secondary button\_text" type="submit" >Login</button>

</div>

</form>

</div>

</div>

<?php require\_once('../../bootstrap/script.php');?>

<script src="script.js" defer ></script>

</body>

</html>

login.js

*document*.addEventListener('DOMContentLoaded', (e) => {

e.preventDefault();

const loginParams = {

form: *document*.getElementById('form\_login')

};

function loginErrorReset(){ *document*.getElementById('err\_username\_login').textContent = ''; *document*.getElementById('err\_password\_login').textContent = '';

}

function errorsLog(code){ switch (code){

case 1:

case 2:

return "err\_username\_login";

case 3:

case 4:

return "err\_password\_login";

}

}

loginParams.form.onsubmit = async (e) => { e.preventDefault();

const formData = new FormData(loginParams.form);

const response = await data(formData).then(response => response.json()); loginErrorReset();

if(response.errors?.length >= 1){

for (const error of response.errors) {

const className = errorsLog(error.code);

*document*.getElementById(className).textContent = error.message;

}

}

else{

loginParams.form.reset(); loginErrorReset(); *window*.location.href = response.url;

}

}

async function data(data){

const url = "http://localhost:63342/furniture\_factory/auth/login/validate.php";

return await fetch(url, { method: 'POST', body: data

});

}

});

register.php

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../bootstrap/style.php');?>

<link rel="stylesheet" href="./style.css">

<title>Register</title>

</head>

<body>

<div class="row mx-2">

<div id="card\_register" class="card col-md-6 offset-md-3 px-0 mt-5" >

<form id="form\_register">

<div class="card-header d-flex justify-content-between">

<div class="modal-title text-title">Register</div>

<a href="http://localhost:63342/furniture\_factory/auth/login/login.php" class="text text- decoration-none text-custom" >Login</a>

</div>

<div class="card-body">

<div class="my-3" >

<label for="username">Username</label>

<input autocomplete="false" type="text" name="username" id="username" placeholder="Username" class="form-control text-custom" autofocus>

<span id="err\_username\_register" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="firstname">FirstName</label>

<input autocomplete="false" type="text" name="firstname" id="firstname" placeholder="Firstname" class="form-control text-custom">

<span id="err\_firstname\_register" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="lastname">LastName</label>

<input autocomplete="false" type="text" name="lastname" id="lastname" placeholder="Lastname" class="form-control text-custom">

<span id="err\_lastname\_register" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="email">Email</label>

<input autocomplete="false" type="text" name="email" id="email" placeholder="Email" class="form-control text-custom">

<span id="err\_email\_register" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="password">Password</label>

<input autocomplete="false" type="password" name="password" id="password" placeholder="Password" class="form-control text-custom">

<span id="err\_password\_register" class="text-custom text text-danger"></span>

</div>

<div class="my-3" >

<label for="re-password">Repeat Password</label>

<input autocomplete="false" type="password" name="re-password" id="re-password" placeholder="Repeat Password" class="form-control text-custom">

<span id="err\_re-password\_register" class="text-custom text text-danger"></span>

</div>

</div>

<div class="card-footer text-end">

<button class="btn btn-secondary button\_text" type="submit" >Register</button>

</div>

</form>

</div>

</div>

<?php require\_once('../../bootstrap/script.php');?>

<script src="script.js" defer ></script>

</body>

</html>

register.js

*document*.addEventListener('DOMContentLoaded', (e) => {

e.preventDefault();

const registerParams = {

form: *document*.getElementById('form\_register')

};

function registerErrorReset(){ *document*.getElementById('err\_username\_register').textContent = ''; *document*.getElementById('err\_firstname\_register').textContent = ''; *document*.getElementById('err\_lastname\_register').textContent = ''; *document*.getElementById('err\_email\_register').textContent = ''; *document*.getElementById('err\_password\_register').textContent = ''; *document*.getElementById('err\_re-password\_register').textContent = '';

}

function errorsLog(code){ switch (code){

case 1:

case 2:

return "err\_username\_register"; case 3:

return "err\_firstname\_register"; case 4:

return "err\_lastname\_register"; case 5:

case 6:

return "err\_email\_register";

case 7:

return "err\_password\_register"; case 8:

return "err\_re-password\_register";

}

}

registerParams.form.onsubmit = async (e) => { e.preventDefault();

const formData = new FormData(registerParams.form);

const response = await data(formData).then(response => response.json());

registerErrorReset();

if(response.errors?.length >= 1){

for (const error of response.errors) {

const className = errorsLog(error.code);

*document*.getElementById(className).textContent = error.message;

}

}

else{

registerParams.form.reset(); registerErrorReset();

*window*.location.href = response.url;

}

}

async function data(data){

const url = "http://localhost:63342/furniture\_factory/auth/register/validate.php"; return await fetch(url, {

method: 'POST', body: data

});

}

});

validate.php

<?php

header('Access-Control-Allow-Origin: http://localhost:63342/\*'); header('Access-Control-Allow-Methods: POST'); header('Access-Control-Allow-Headers: Content-Type');

require\_once('../../sqlMain.php');

$sqlMain = new MainSql();

if($\_SERVER['REQUEST\_METHOD'] === 'POST'){

$request = $\_POST;

$errors = [];

if(empty($request['username'])){

$errors[] = ['code' => 1, 'message' => "You don't fill username field!"];

}

if(!empty($request['username'])){

$checkUsername = $sqlMain->checkUsernameClients($request['username']) ||

$sqlMain->checkUsernameAdmin($request['username']); if(!$checkUsername){

$errors[] = ['code' => 2, 'message' => "This username was exists!"];

}

}

if(empty($request['firstname'])){

$errors[] = ['code' => 3, 'message' => "You don't fill firstname field!"];

}

if(empty($request['lastname'])){

$errors[] = ['code' => 4, 'message' => "You don't fill secondname field!"];

}

if(empty($request['email'])){

$errors[] = ['code' => 5, 'message' => "You don't fill email field!"];

}

if(!empty($request['email']) && !filter\_var($request['email'], *FILTER\_VALIDATE\_EMAIL*)){

$errors[] = ['code' => 6, 'message' => "Email is not valid!"];

}

if(empty($request['password'])){

$errors[] = ['code' => 7, 'message' => "You don't fill password field!"];

}

if(empty($request['re-password'])){

$errors[] = ['code' => 8, 'message' => "You don't fill re-password field!"];

}

if(!empty($request['password']) && !empty($request['re-password'])){

if($request['password'] !== $request['re-password']){

$errors[] = ['code' => 8, 'message' => "Passwords is not equal!"];

}

}

if(!empty($errors)){

echo json\_encode(['status' => false, 'errors' => $errors]);

die();

}

else{

if($res = $sqlMain->addClient($request)){ echo json\_encode(['status' => true, 'url' =>

"http://localhost:63342/furniture\_factory/auth/login/login.php"]);

}

}

}

order.php

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../../bootstrap/style.php');?>

<?php require\_once('../../../bootstrap/fontawesome.php');?>

<link rel="stylesheet" href="./style.css">

<title>Order</title>

</head>

<body>

<?php

require\_once('../../../sqlMain.php');

$sqlMain = new MainSql();

?>

<div class="container-md position-relative">

<div style="position: relative; z-index: 1;">

<div id="alert\_success" class="position-fixed p-1 bottom-0 left-0 mb-3 d-flex" style="transform: translateY(150%); transition: transform .2s ease-in;">

<span class="card d-flex align-items-center justify-content-center py-3 px-3 border-end-0 rounded-end-0 bg-success text-custom">

<i class="fa-solid fa-check"></i>

</span>

<span class="card align-items-center justify-content-center py-3 px-3 rounded-start-0 text- custom" id="success\_message">

</span>

</div>

</div>

<div class="d-flex justify-content-between align-items-center my-3">

<h2 class="text-title">Welcome in order, <?php echo $sqlMain->getUsername(); ?></h2>

<button class="btn btn-secondary" type="button" id="back\_btn">Back</button>

</div>

<ul class="nav nav-tabs" role="tablist">

<li class="nav-item">

<a id="search\_button" class="nav-link text text-black active text-custom" data-bs- toggle="tab" href="#search\_order">Search Order</a>

</li>

<li class="nav-item">

<a id="create\_button" class="nav-link text text-black text-custom" data-bs-toggle="tab" href="#create\_order">Create Order</a>

</li>

</ul>

<div class="tab-content">

<div id="create\_order" class="tab-pane fade">

<div class="card border-top-0 rounded-top-0">

<form id="form\_create">

<div class="card-header">

<div class="text-title">Create</div>

</div>

<div class="card-body">

<div class="my-2">

<label class="text-custom" for="create\_username">Username</label>

<input type="text" class="form-control shadow-none text-custom" name="create\_username" id="create\_username" autofocus autocomplete="false">

<span id="err\_create\_username" class="text text-danger text-custom"></span>

</div>

<div class="my-2">

<label class="text-custom" for="create\_order\_name">Order Name</label>

<input type="text" class="form-control shadow-none text-custom" name="create\_order\_name" id="create\_order\_name" autocomplete="false">

<span id="err\_create\_order\_name" class="text text-danger text-custom"></span>

</div>

<div class="my-2">

<label class="text-custom" for="create\_order\_description">Order

Description</label>

<textarea class="form-control shadow-none text-custom"

name="create\_order\_description" id="create\_order\_description" autocomplete="false"></textarea>

<span id="err\_create\_order\_description" class="text text-danger text-

custom"></span>

</div>

<div class="my-2" >

<label class="text-custom" for="create\_perform\_to">Perform To</label>

<input type="time" class="form-control shadow-none text-custom"

name="create\_perform\_to" id="create\_perform\_to" autocomplete="false">

<span id="err\_create\_perform\_to" class="text text-danger text-custom"></span>

</div>

</div>

<div class="card-footer text-end">

<button class="btn btn-secondary button\_text" type="submit">Create</button>

</div>

</form>

</div>

</div>

<div id="search\_order" class="tab-pane fade show active">

<div class="card border-top-0 rounded-top-0">

<form id="form\_search">

<div class="card-header">

<div class="text-title">Search</div>

</div>

<div class="card-body">

<div class="my-2">

<label class="text-custom" for="search\_username">Username</label>

<input type="text" class="form-control shadow-none text-custom" name="search\_username" id="search\_username" autofocus autocomplete="false">

<span id="err\_search\_username" class="text text-danger text-custom"></span>

</div>

</div>

<div class="card-footer text-end">

<button class="btn btn-secondary button\_text" type="submit">Search</button>

</div>

</form>

<div class="none px-2" id="table">

<table class="table table-bordered text\_table" >

<thead>

<tr>

<th>Username</th>

<th>Order name</th>

<th>Order Description</th>

<th>Perform To</th>

<th>Options</th>

</tr>

</thead>

<tbody id="body\_search">

</tbody>

</table>

</div>

</div>

</div>

</div>

<div class="modal fade" id="update\_modal\_order" >

<div class="modal-dialog">

<div class="modal-content">

<form class="form" id="form\_update\_order">

<div class="modal-header">

<div class="modal-title text-title">Update</div>

<button id="close-update-header" class="btn btn-close button\_text" data-bs- dismiss="modal" type="button"></button>

</div>

<div class="modal-body">

<div class="my-2">

<label class="text-custom" for="update\_order\_name">Order Name</label>

<input type="text" class="form-control shadow-none text-custom" name="update\_order\_name" id="update\_order\_name" autocomplete="false">

<span id="err\_update\_order\_name" class="text text-danger text-custom"></span>

</div>

<div class="my-2">

<label class="text-custom" for="update\_order\_description">Order

Description</label>

<textarea class="form-control shadow-none text-custom"

name="update\_order\_description" id="update\_order\_description" autocomplete="false"></textarea>

<span id="err\_update\_order\_description" class="text text-danger text-

custom"></span>

</div>

<div class="my-2" >

<label class="text-custom" for="update\_perform\_to">Perform To</label>

<input type="time" class="form-control shadow-none text-custom" name="update\_perform\_to" id="update\_perform\_to" autocomplete="false">

<span id="err\_update\_perform\_to" class="text text-danger text-custom"></span>

</div>

</div>

<div class="modal-footer">

<button id="close-update-footer" class="btn btn-danger button\_text" type="button" data-bs-dismiss="modal" >Cancel</button>

<button type="submit" class="btn btn-primary" >Update</button>

</div>

</form>

</div>

</div>

</div>

<div class="modal fade" id="delete\_modal\_order" >

<div class="modal-dialog">

<div class="modal-content">

<form id="form\_delete\_order" class="form">

<div class="modal-header">

<div class="modal-title text-title">Delete</div>

<button id="delete-close-header" type="button" class="btn btn-close" data-bs- dismiss="modal"></button>

</div>

<div class="modal-body">

<span>

Are you sure you want delete Order: <b id="delete\_text"></b> ?

</span>

</div>

<div class="modal-footer">

<button id="delete-close-footer" type="button" class="btn btn-danger" data-bs- dismiss="modal">Cancel</button>

<button type="submit" class="btn btn-primary">Delete</button>

</div>

</form>

</div>

</div>

</div>

</div>

<?php require\_once('../../../bootstrap/script.php');?>

<script src="script.js" defer ></script>

</body>

</html>

order.js

*document*.addEventListener('DOMContentLoaded', (e) => { e.preventDefault();

*document*.getElementById('back\_btn').onclick = (e) => { e.preventDefault();

*window*.location.href = '../menu/index.php';

}

const searchParams = {

form: *document*.getElementById('form\_search')

}

const createParams = {

form: *document*.getElementById('form\_create')

}

const success\_alert = {

success: *document*.getElementById('alert\_success'), message\_success: *document*.getElementById('success\_message')

}

*document*.getElementById('search\_button').onclick = (e) => {

setTimeout(() => { errorSearchFormReset(); searchParams.form.reset();

if(!*document*.getElementById('table').classList.contains('none')){

*document*.getElementById('table').classList.add('none');

}

}, 500);

}

*document*.getElementById('create\_button').onclick = (e) => { setTimeout(() => {

errorCreateFormReset(); createParams.form.reset();

if(!*document*.getElementById('table').classList.contains('none')){

*document*.getElementById('table').classList.add('none');

}

}, 500);

}

// createFormErrors

function errorCreateFormReset(){

*document*.getElementById('err\_create\_username').textContent = '';

*document*.getElementById('err\_create\_order\_name').textContent = ''; *document*.getElementById('err\_create\_order\_description').textContent = ''; *document*.getElementById('err\_create\_perform\_to').textContent = '';

}

function createFormLog(code){ switch (code){

case 1:

case 2:

return 'err\_create\_username'; case 3:

return 'err\_create\_order\_name'; case 4:

return 'err\_create\_order\_description'; case 5:

return 'err\_create\_perform\_to';

}

}

// searchFormErrors

function errorSearchFormReset(){

*document*.getElementById('err\_search\_username').textContent = '';

}

function searchFormLog(code){

switch (code){ case 1:

case 2:

return 'err\_search\_username';

}

}

// search

searchParams.form.onsubmit = async (e) => { e.preventDefault();

const formData = new FormData(searchParams.form); formData.append('action', 'search\_order');

const response = await data(formData).then(response => response.json());

errorSearchFormReset();

if(response.errors?.length >= 1){

for (const error of response.errors) {

const className = searchFormLog(error.code);

*document*.getElementById(className).textContent = error.message;

}

}

else{

errorSearchFormReset(); searchParams.form.reset();

if(response.data.empty\_message){

renderMessage(response.data.empty\_message);

}

else{

renderTable(response.data);

}

}

}

function renderTable(data){ if(data){

*document*.getElementById('body\_search').innerHTML = ''; for (const row of data) {

const tableOrders = `

<tr data-id="${row.id}">

<td class="order\_username">${row.username}</td>

<td class="order\_name">${row.order\_name}</td>

<td class="order\_description">${row.order\_description}</td>

<td class="perform\_to">${row.perform\_to}</td>

<td>

<div class="d-flex align-items-center justify-content-center flex-row">

<button data-delete="${row.id}" type="button" class="border border-end-0 rounded rounded-end-0 delete-btn">

<i class='fa fa-trash' style="pointer-events: none;"></i>

</button>

<button data-update="${row.id}" type="button" class="border rounded rounded-start-0

update-btn">

<i class="fa fa-pen"></i>

</button>

</div>

</td>

</tr>

`;

*document*.getElementById('body\_search').insertAdjacentHTML('beforeend', tableOrders); const btnUpdate = *document*.querySelector(`#body\_search tr[data-id="${parseInt(row.id)}"]

button[data-update="${parseInt(row.id)}"]`); updateBindContext(btnUpdate);

const btnDelete = *document*.querySelector(`#body\_search tr[data-id="${parseInt(row.id)}"] button[data-delete="${parseInt(row.id)}"]`);

deleteBindContext(btnDelete);

}

*document*.getElementById('search\_table\_empty')?.remove();

*document*.getElementById('table').classList.remove('none');

}

}

function renderMessage(message){ const messageTable = `

<tr id="search\_table\_empty">

<td colspan="5" class="text-center text-title">${message}</td>

</tr>

`; if(!*document*.getElementById('body\_search').getElementById('search\_table\_empty')) {

*document*.getElementById('body\_search').insertAdjacentHTML('beforeend', messageTable);

}

*document*.getElementById('table').classList.remove('none');

}

// create

createParams.form.onsubmit = async (e) => { e.preventDefault();

const formData = new FormData(createParams.form); formData.append('action', 'create\_order');

const response = await data(formData).then(response => response.json()); errorCreateFormReset();

if(response.errors?.length >= 1){

for (const error of response.errors){

const className = createFormLog(error.code);

*document*.getElementById(className).textContent = error.message;

}

}

else{

createParams.form.reset(); errorCreateFormReset();

success\_alert.message\_success.textContent = 'You successfully add order!'; success\_alert.success.style.setProperty('transform', 'translateY(0%)'); setTimeout(() => {

success\_alert.success.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

// actions update, delete

const updateModal = new bootstrap.Modal(*document*.getElementById('update\_modal\_order')); const deleteModal = new bootstrap.Modal(*document*.getElementById('delete\_modal\_order'));

const updateParams = {

form: *document*.getElementById('form\_update\_order'), footerClose: *document*.getElementById('close-update-footer'), headerClose: *document*.getElementById('close-update-header'), id\_update: null

}

const deleteParams = {

form: *document*.getElementById('form\_delete\_order'),

text: *document*.getElementById('delete\_text'),

footerClose: *document*.getElementById('delete-close-footer'), headerClose: *document*.getElementById('delete-close-header'), id\_delete: null

}

const defaultParams = {

updateButtons: *document*.querySelectorAll('tr .update-btn'),

deleteButtons: *document*.querySelectorAll('tr .delete-btn')

}

// update

updateModal.\_element.onclick = (e) => { if(!e.target.closest('#form\_update\_order')){

updateErrorsClear();

updateParams.form.reset();

}

}

updateParams.footerClose.onclick = (e) => { updateErrorsClear(); updateParams.form.reset();

}

updateParams.headerClose.onclick = (e) => { updateErrorsClear(); updateParams.form.reset();

}

function updateErrorsClear(){ *document*.getElementById('err\_update\_order\_name').textContent = ''; *document*.getElementById('err\_update\_order\_description').textContent = '';

*document*.getElementById('err\_update\_perform\_to').textContent = '';

}

function errUpdateLog(code){ switch(code){

case 1:

return 'err\_update\_order\_name'; case 2:

return 'err\_update\_order\_description';

case 3:

return 'err\_update\_perform\_to';

}

}

defaultParams.updateButtons.forEach(btn => {

updateBindContext(btn);

});

function updateBindContext(btn){ if(btn){

btn.onclick = (e) => { e.preventDefault();

const getId = parseInt(btn.getAttribute("data-update"));

const trRow = *document*.querySelector(`#body\_search tr[data-id="${getId}"]`);

if(trRow){ updateParams.id\_update = getId;

updateParams.form.elements['update\_order\_name'].value = trRow.querySelector('.order\_name').textContent;

updateParams.form.elements['update\_order\_description'].value =

trRow.querySelector('.order\_description').textContent;

updateParams.form.elements['update\_perform\_to'].value = trRow.querySelector('.perform\_to').textContent;

updateModal.show();

}

}

}

}

updateParams.form.onsubmit = async (e) => { e.preventDefault();

const formData = new FormData(updateParams.form); formData.append('action', 'update\_order'); formData.append('id', updateParams.id\_update);

const response = await data(formData).then(response => response.json()); updateErrorsClear();

if(response.errors?.length >= 1){

for (const error of response.errors) {

const className = errUpdateLog(error.code);

*document*.getElementById(className).textContent = error.message;

}

}

else{

if(response.data){ updateHtml(response.data);

updateParams.footerClose.click();

updateParams.form.reset();

updateErrorsClear();

success\_alert.success.style.setProperty('transform', 'translateY(0%)'); success\_alert.message\_success.textContent = 'You are successfully updated order!'; setTimeout(() => {

success\_alert.success.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

}

function updateHtml(data){ if(data){

const id = parseInt(data.id);

const trRow = *document*.querySelector(`#body\_search tr[data-id="${id}"]`);

if(trRow){

trRow.querySelector('.order\_name').textContent = data.order\_name; trRow.querySelector('.order\_description').textContent = data.order\_description; trRow.querySelector('.perform\_to').textContent = data.perform\_to;

}

}

}

// delete order defaultParams.deleteButtons.forEach(btn => {

deleteBindContext(btn);

});

function deleteBindContext(btn){ if(btn){

btn.onclick = (e) => { e.preventDefault();

const getId = parseInt(e.target.getAttribute('data-delete'));

const trRow = *document*.querySelector(`#body\_search tr[data-id="${getId}"]`);

if(trRow){

deleteParams.id\_delete = getId;

deleteParams.text.textContent = trRow.querySelector('.order\_name').textContent; deleteModal.show();

}

}

}

}

deleteParams.form.onsubmit = async (e) => {

e.preventDefault();

const dataObj = {id: deleteParams.id\_delete, action: 'delete'};

const response = await data(*Object*.entries(dataObj)).then(response => response.json()); if(response.status){

*document*.querySelector(`#body\_search tr[data-id="${deleteParams.id\_delete}"]`).remove();

deleteParams.footerClose.click(); deleteParams.form.reset();

success\_alert.success.style.setProperty('transform', 'translateY(0%)'); success\_alert.message\_success.textContent = 'You are successfully deleted order!'; setTimeout(() => {

success\_alert.success.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

async function data(data){

const url = "http://localhost:63342/furniture\_factory/view/mainPage/order/validate.php"; return await fetch(url, {

method: 'POST',

body: *JSON*.stringify(*Object*.fromEntries(data)), headers: {

'Content-Type': 'application/json'

}

});

}

});

(order) validate.php

<?php

header('Access-Control-Allow-Origin: http://localhost:63342/\*'); header('Access-Control-Allow-Methods: POST'); header('Access-Control-Allow-Headers: Content-Type'); header('Content-Type: application/json');

require\_once('../../../sqlMain.php');

$sqlMain = new MainSql();

if($\_SERVER['REQUEST\_METHOD'] === 'POST'){

$request = file\_get\_contents('php://input');

$request = json\_decode($request, true);

if($request['action'] === 'search\_order'){

$errors = [];

if(empty($request['search\_username'])){

$errors[] = ['code' => 1, 'message' => 'You must fill username field!'];

}

$current\_username = $sqlMain->getUsername();

if(!empty($request['search\_username']) && $request['search\_username'] !== $current\_username){

$errors[] = ['code' => 2, 'message' => 'Username is wrong!'];

}

if(!empty($errors)){

echo json\_encode(['status' => false, 'errors' => $errors]);

die();

}

else{

if($res = $sqlMain->getOrdersFor($request['search\_username'])){ echo json\_encode(['status' => true, 'data' => $res]);

die();

}

}

}

if($request['action'] === 'create\_order'){

$errors = [];

if(empty($request['create\_username'])){

$errors[] = ['code' => 1, 'message' => 'You must fill username field!'];

}

$current\_username = $sqlMain->getUsername();

if(!empty($request['create\_username']) && $request['create\_username'] !== $current\_username){

$errors[] = ['code' => 2, 'message' => 'This client is not exists!'];

}

if(empty($request['create\_order\_name'])){

$errors[] = ['code' => 3, 'message' => 'You must fill order name field!'];

}

if(empty(htmlspecialchars($request['create\_order\_description']))){

$errors[] = ['code' => 4, 'message' => 'You must fill order description field!'];

}

if(empty($request['create\_perform\_to'])){

$errors[] = ['code' => 5, 'message' => 'You must fill order perform to, field!'];

}

if(!empty($errors)){

echo json\_encode(['status' => false, 'errors' => $errors]); die();

}

else{

if($res = $sqlMain->addOrder($request)){ echo json\_encode(['status' => true]); die();

}

}

}

if($request['action'] === 'update\_order'){

$errors = []; if(empty($request['update\_order\_name'])){

$errors[] = ['code' => 1, 'message' => 'You must fill order name field!'];

}

if(empty(htmlspecialchars($request['update\_order\_description']))){

$errors[] = ['code' => 2, 'message' => 'You must fill order description field!'];

}

if(empty($request['update\_perform\_to'])){

$errors[] = ['code' => 3, 'message' => 'You must fill order perform to, field!'];

}

if(!empty($errors)){

echo json\_encode(['status' => false, 'errors' => $errors]); die();

}

else{

if($res = $sqlMain->updateOrder($request)){

echo json\_encode(['status' => true, 'data' => $res]);

die();

}

}

}

if($request['action'] === 'delete'){

if($res = $sqlMain->deleteOrder($request['id'])){

echo json\_encode(['status' => true]);

}

}

}

menu.php

<?php

session\_start();

$userStatus = isset($\_GET['admin']) ? 'admin' : (isset($\_GET['client']) ? 'client' : 'guest');

$username = null;

if ($userStatus === 'admin' && isset($\_GET['admin'])) {

$username = $\_GET['admin'];

} elseif ($userStatus === 'client' && isset($\_GET['client'])) {

$username = $\_GET['client'];

}

if ($username && $userStatus) {

$\_SESSION['username'] = $username;

$\_SESSION['status'] = $userStatus;

}

?>

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="./style.css" >

<?php require\_once('../../../bootstrap/style.php'); ?>

<title>Furniture Factory</title>

</head>

<body class="gray-custom">

<div class="container-md p-0">

<div class="card border-0 mt-3 px-2 shadow" >

<nav class="navbar navbar-expand-md px-2 rounded">

<a class="navbar-brand text-title text-decoration-none" href="#">

FurnitureCo.

<h4 class="button\_text text-decoration-none m-0">Welcome, <?php echo

$\_SESSION['username']; ?>!</h4>

</a>

<button class="navbar-toggler text-custom shadow-none" type="button" data-bs- toggle="collapse" data-bs-target="#menu" aria-controls="menu" aria-expanded="false" aria- label="Toggle navigation">

<span class="navbar-toggler-icon text-custom"></span>

</button>

<div class="collapse navbar-collapse" id="menu">

<ul class="navbar-nav ms-auto gap-2">

<li class="nav-item">

<button style="background-color: #e8e8e8" type="button" class="btn button\_text rounded-5" id="work\_staff\_btn" >Our Work Staff</button>

</li>

<li class="nav-item <?php echo $\_SESSION['status'] === 'admin' ? 'd-none' : ''; ?>">

<button style="background-color: #e8e8e8" type="button" id="order\_btn" class="btn button\_text rounded-5">Order</button>

</li>

<li class="nav-item">

<button type="button" id='quit\_btn' class="btn btn-danger text-button">Quit</button>

</li>

</ul>

</div>

</nav>

</div>

<div class="modal fade" id="confirm\_quit" >

<div class="modal-dialog">

<div class="modal-content">

<form id="form\_quit">

<div class="modal-header">

<div class="modal-title text-title">Quit</div>

<button class="btn btn-close" data-bs-dismiss="modal" type="button"></button>

</div>

<div class="modal-body">

<span class="button\_text">

Are you sure you want quit <b><?php echo $\_SESSION['username']; ?></b>?

</span>

</div>

<div class="modal-footer">

<button class="btn btn-secondary" type="button" data-bs-

dismiss="modal">Cancel</button>

<button class="btn btn-danger" type="submit">Quit</button>

</div>

</form>

</div>

</div>

</div>

<div id="about-factory" class="about\_factory">

<?php require\_once('../about/index.php'); ?>

</div>

</div>

<?php require\_once('../../../bootstrap/script.php'); ?>

<script src="./script.js" defer ></script>

</body>

</html>

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-

scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../../bootstrap/style.php'); ?>

<link rel="stylesheet" href="./style.css">

<title>About Factory</title>

</head>

<body>

<?php

require\_once('../../../sqlMain.php');

$sqlMain = new MainSql();

?>

<div class="container-md p-0 mt-4">

<div class="border-0 d-flex justify-content-between align-items-center flex-row p-0 mt-5" >

<div class="d-flex flex-column me-4" >

<h2 class="m">About Us</h2>

<div class="button\_text">

FurnitureCo is a leading manufacturer of high-quality furniture, renowned for

its exceptional craftsmanship, innovative designs, and commitment to sustainability. Established in 2000, our factory is located in the heart of City A, where we combine

traditional beauty and

woodworking techniques with modern technology to create timeless pieces that enhance the functionality of any space.

</div>

</div>

<div style="object-fit: contain;">

<img style="width: 100%; height: 100%;" src="../../mainPage/about/images/main.jpeg"

alt="Main Image">

</div>

</div>

<div class="border-0 d-flex justify-content-between align-items-center flex-row p-0 mt-5" >

<div style="object-fit: contain;" >

<img style="width: 100%; height: 100%;" src="../../mainPage/about/images/white\_sofa.jpg" alt="Sofa Image">

</div>

<div class="d-flex flex-column ms-5">

<h2 class="m">Crafting High-Quality Furniture with Personalized Room Design Services</h2>

<div class="button\_text">

FurnitureCo is a leading manufacturer of high-quality furniture, renowned for its exceptional craftsmanship, innovative designs, and commitment to sustainability. Established in 2000, our factory is located in the heart of City A, where we combine traditional woodworking techniques with modern

technology to create timeless pieces that enhance the beauty and functionality of any space. Additionally, we offer a unique service that allows you to create your own room design, ensuring that your furniture perfectly complements your personal style and needs. Our team of experts is here to help you bring your vision to life, from initial concept to final installation.

</div>

</div>

</div>

<div class="border-0 d-flex justify-content-between align-items-center flex-row p-0 mt-5" >

<div class="d-flex flex-column me-4" >

<div class="button\_text">

FurnitureCo is a leading manufacturer of premium furniture, celebrated for its superior craftsmanship, cutting-edge designs, and dedication to sustainable practices. Since our establishment in 2000, we have been nestled in the heart of City A, merging age-old woodworking techniques with contemporary technology to produce timeless pieces that elevate the aesthetics and usability of any environment. We also empower you to create your own room design, tailoring each piece to fit your unique style and requirements. Our skilled professionals will guide you through every step of the process, ensuring your dream space becomes a reality.

</div>

</div>

<div>

<img style="width: 100%; height: 100%;" src="../../mainPage/about/images/cream\_sofa.jpg" alt="Cream Sofa Image">

</div>

</div>

<div class="border-0 d-flex justify-content-between align-items-center flex-row p-0 mt-5" >

<div>

<img style="width: 100%; height: 100%;" src="../../mainPage/about/images/room.jpg" alt="Cream Sofa Image">

</div>

<div class="d-flex flex-column ms-5" >

<div class="button\_text">

At FurnitureCo, we pride ourselves on being a premier furniture manufacturer known for our exceptional craftsmanship, innovative design ethos, and strong commitment to sustainability. Founded in 2000, our factory in the heart of City A harmonizes classic woodworking techniques with state-of-the-art technology, producing enduring pieces that beautifully enhance the functionality of any setting. We also offer a unique service that allows you to create your own room design, ensuring that each piece is tailored to your specific tastes and requirements. Our expert team is here to support you every step of the way, transforming your design ideas into reality.

</div>

</div>

</div>

<div class="d-flex flex-column my-2 my-5 text-center mt-5">

<span class="text-title">About Our Manufacturing Facility</span>

<div class="image-container position-relative">

<img class="image" src="../../mainPage/about/images/desk.avif" alt="Info Image">

<table class="text\_table table table-bordered my-1 overlay-table">

<thead>

<tr>

<th>Factory Name</th>

<th>Factory Description</th>

<th>Factory Location</th>

<th>Data Established</th>

<th>Number Of Employers</th>

<th>Annual Revenue</th>

</tr>

</thead>

<tbody>

<?php

$array = $sqlMain->getData('factory'); if(count($array)):

foreach ($array as $row):

?>

<tr>

<td><?php echo $row['factory\_name']; ?></td>

<td><?php echo $row['factory\_description']; ?></td>

<td><?php echo $row['factory\_location']; ?></td>

<td><?php echo $row['established\_date']; ?></td>

<td>

<?php

echo $sqlMain->getSumWorkers();

?>

</td>

<td><?php echo $row['annual\_revenue']; ?></td>

</tr>

<?php endforeach;

else:

?>

<tr>

<td colspan="6" class="text-title text-center" >Table is Empty!</td>

</tr>

<?php endif;

?>

</tbody>

</table>

</div>

</div>

<style>

.image-container { position: relative; width: 100%; height: 65vh;

}

.image {

width: 100%;

height: 100%; object-fit: contain;

}

.overlay-table { position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background: rgba(255, 255, 255, 0.8);

width: 40%;

border-collapse: collapse;

}

.overlay-table th, .overlay-table td { padding: 10px;

border: 1px solid #ddd; text-align: left;

}

.overlay-table th {

background-color: #f2f2f2;

}

@media screen and (max-width: 600px){

.overlay-table{ width: 10%; font-size: 9px;

}

.overlay-table th, .overlay-table td {

padding: 2px;

}

}

@media screen and (max-width: 450px){

.overlay-table{ width: 5%; font-size: 7px;

}

.overlay-table th, .overlay-table td { padding: 1px;

}

}

@media screen and (max-width: 360px){

.overlay-table{ width: 3%; font-size: 5px;

}

.overlay-table th, .overlay-table td { padding: 1px;

}

}

</style>

<?php require\_once('../../../bootstrap/script.php'); ?>

</body>

</html>

main.php departments menu

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../../../bootstrap/style.php'); ?>

<link rel="stylesheet" href="./style.css">

<title>Departments</title>

</head>

<body>

<div class="container-md p-0 d-flex flex-column px-5 gap-3">

<div class="d-flex justify-content-between my-3" >

<h2 class="text-title" >Departments</h2>

<button class="btn btn-secondary button\_text" id="back\_btn">Back</button>

</div>

<div class="d-flex justify-content-between">

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="./images/ProductionDepartment.jpg" alt="Production

Department">

Staff</a>

<div class="card-body">

<h5 class="card-title text-title">Production Department</h5>

<p class="card-text text-custom">

The Production Department at FurnitureCo.

crafts high-quality furniture with precision and care.

Skilled workers and advanced machinery transform raw materials into durable, beautiful pieces, ensuring excellence in every product.

</p>

<a id="production\_department" href="#" class="btn btn-secondary button\_text">View

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="./images/DesignDepartment.jpg" alt="Design Department">

<div class="card-body">

<h5 class="card-title text-title">Design Department</h5>

<p class="card-text text-custom">

The Design Department at FurnitureCo. innovates and creates stylish furniture concepts. Talented designers blend aesthetics with functionality,

producing unique designs that cater to diverse tastes and enhance living spaces.

</p>

<a id="design\_department" href="#" class="btn btn-secondary button\_text">View Staff</a>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="./images/ItDepartment.jpg" alt="It Department">

<div class="card-body">

<h5 class="card-title text-title">It Department</h5>

<p class="card-text text-custom">

The IT Department at FurnitureCo. ensures smooth technological operations. They manage software, hardware, and network infrastructure,

supporting efficient processes across all departments and enhancing productivity and customer experience.

</p>

<a id="it\_department" href="#" class="btn btn-secondary button\_text">View Staff</a>

</div>

</div>

</div>

<div class="mt-3" style="display: flex; justify-content: space-around;">

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="./images/HrDepartment.jpg" alt="Hr Department">

<div class="card-body">

<h5 class="card-title text-title">Hr Department</h5>

<p class="card-text text-custom">

The HR Department at FurnitureCo. is dedicated to managing and supporting the company's most valuable asset: its people. They handle recruitment, training, employee relations, and benefits, ensuring a positive and productive work environment for all staff.

</p>

<a id="hr\_department" href="#" class="btn btn-secondary button\_text">View Staff</a>

</div>

</div>

<div class="card" style="width: 18rem;">

<img class="card-img-top" src="./images/QualityControlDepartment.jpg" alt="Quality Control

Department">

<div class="card-body">

Staff</a>

<h5 class="card-title text-title">Quality Control Department</h5>

<p class="card-text text-custom">

The Quality Control Department at FurnitureCo. conducts thorough inspections and tests to ensure every product meets our high standards

\for quality and safety. They play a crucial role in maintaining the excellence and reliability of our furniture.

</p>

<a id="quality\_control\_department" href="#" class="btn btn-secondary button\_text">View

</div>

</div>

</div>

</div>

<?php require\_once('../../../../bootstrap/script.php'); ?>

<script src="./script.js" defer ></script>

</body>

</html>

design.php тобто я покажу роботу з 1 таблицею відділень так як вони схожі за логікою

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum- scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<?php require\_once('../../../../bootstrap/style.php'); ?>

<?php require\_once('../../../../bootstrap/fontawesome.php'); ?>

<link rel="stylesheet" href="./style.css">

<title>Design Department</title>

</head>

<body>

<?php

require\_once('../../../../sqlMain.php');

$sqlMain = new MainSql();

?>

<div class="container-md my-2 py-2">

<div class="" style="position: relative;">

<div id="alert\_success" class="position-fixed d-flex flex-row bottom-0 left-0 py-3" style="transform: translateY(150%); transition: transform .5s ease-in-out;">

<span class="d-flex align-items-center justify-content-center card bg-success rounded rounded- end-0 border-end-0 px-3 py-3 text-custom">

<i class="fa-solid fa-check"></i>

</span>

<span id="success\_message" class="card rounded rounded-start-0 d-flex align-items-center p-3 text-custom"></span>

</div>

</div>

<div class="" style="position: relative;">

<div id="alert\_warning\_add" class="position-fixed d-flex flex-row bottom-0 left-0 py-3" style="transform: translateY(150%); transition: transform .5s ease-in-out;">

<span class="d-flex align-items-center justify-content-center card bg-warning rounded

rounded-end-0 border-end-0 px-3 py-3 text-custom">

<i class="fa-solid fa-circle-exclamation"></i>

</span>

<span class="card rounded rounded-start-0 d-flex align-items-center p-3 text-custom">You don't have access to add!</span>

</div>

</div>

<div class="" style="position: relative;">

<div id="alert\_warning\_update" class="position-fixed d-flex flex-row bottom-0 left-0 py-3" style="transform: translateY(150%); transition: transform .5s ease-in-out;">

<span class="d-flex align-items-center justify-content-center card bg-warning rounded rounded-end-0 border-end-0 px-3 py-3 text-custom">

<i class="fa-solid fa-circle-exclamation"></i>

</span>

<span class="card rounded rounded-start-0 d-flex align-items-center p-3 text-custom">You don't have access to update!</span>

</div>

</div>

<div class="" style="position: relative;">

<div id="alert\_warning\_delete" class="position-fixed d-flex flex-row bottom-0 left-0 py-3" style="transform: translateY(150%); transition: transform .5s ease-in-out;">

<span class="d-flex align-items-center justify-content-center card bg-warning rounded rounded-end-0 border-end-0 px-3 py-3 text-custom">

<i class="fa-solid fa-circle-exclamation"></i>

</span>

<span class="card rounded rounded-start-0 d-flex align-items-center p-3 text-custom">You don't have access to delete!</span>

</div>

</div>

<div class="d-flex justify-content-between" >

<h2 class="text-title m-0" >Design Department</h2>

<button class="btn btn-secondary" id="back\_btn" >Back</button>

</div>

<div class="d-flex flex-row my-3">

<button type="button" class="btn btn-primary rounded-end-0 text-custom" id="add">Add</button>

<input type="search" class="form-control shadow-none rounded-start-0 rounded-end-0 text- custom" id="search\_inp" placeholder="Search by worker full name" >

<button disabled id="search\_btn" type="button" class="btn btn-primary rounded-start-0 text-

custom" >Search</button>

</div>

<table class="table table-bordered text\_table my-3">

<thead>

<tr>

<th>Employee Name</th>

<th>Position</th>

<th>Project Name</th>

<th>Project Description</th>

<th>Start Date</th>

<th>End Date</th>

<th>Salary</th>

<th>Employee Image</th>

<th>Options</th>

</tr>

</thead>

<tbody id="table\_body">

<?php

$array = $sqlMain->getData('design\_development'); if(sizeof($array)):

foreach ($array as $row):

?>

<tr data-id="<?php echo $row['id']; ?>">

<td class="employee\_name"><?php echo $row['employee\_name'];?></td>

<td class="position"><?php echo $row['position'];?></td>

<td class="project\_name"><?php echo $row['project\_name'];?></td>

<td class="project\_description"><?php echo $row['project\_description'];?></td>

<td class="start\_date"><?php echo $row['start\_date'];?></td>

<td class="end\_date"><?php echo $row['end\_date'];?></td>

<td class="salary"><?php echo $row['salary'];?></td>

<td>

<img data-img="<?php echo $row['id']; ?>" src="../../workStaff/design/dd/DD<?php echo (int)$row['id']; ?>.jpg" style="width: 75px; height: 75px; object-fit: cover;" alt="Employee Image">

</td>

<td>

<div class="d-flex align-items-center justify-content-center flex-row">

<button data-delete="<?php echo $row['id']; ?>" type="button" class="border border- end-0 rounded rounded-end-0 delete-btn">

<i class='fa fa-trash' style="pointer-events: none;"></i>

</button>

<button data-update="<?php echo $row['id']; ?>" type="button" class="border rounded rounded-start-0 update-btn">

<i class="fa fa-pen"></i>

</button>

</div>

</td>

</tr>

<?php

endforeach; else:

?>

<tr>

<td colspan="9" class="text-title text-center" >Table is Empty!</td>

</tr>

<?php

endif;

?>

</tbody>

</table>

<div class="modal fade" id="add\_update\_modal">

<div class="modal-dialog">

<div class="modal-content">

<form id="add\_update\_form" enctype="multipart/form-data">

<div class="modal-header">

<div class="modal-title fs-2 text-title" id="title\_modal"></div>

<button id="close-modal-header" type="button" class="btn btn-close shadow-none

button\_text" data-bs-dismiss="modal"></button>

</div>

<div class="modal-body">

<input id="update\_id" type="hidden" value="" name="update\_id">

<div class="my-3" >

<label class="text-custom" for="employee\_name">Employee Name</label>

<input type="text" min="1" class="form-control shadow-none text-custom" name="employee\_name" id="employee\_name" autofocus>

<span id="err\_employee\_name" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="position">Position</label>

<input type="text" class="form-control shadow-none text-custom" name="position"

id="position">

<span id="err\_position" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="project\_name">Project Name</label>

<input type="text" class="form-control shadow-none text-custom" name="project\_name" id="project\_name">

<span id="err\_project\_name" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="project\_description">Project Description</label>

<textarea class="form-control shadow-none text-custom" name="project\_description" id="project\_description"></textarea>

<span id="err\_project\_description" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="start\_data">Start Date</label>

<input type="date" class="form-control shadow-none text-custom" name="start\_date" id="start\_data">

<span id="err\_start\_data" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="end\_data">End Date</label>

<input type="date" class="form-control shadow-none text-custom" name="end\_date"

id="end\_data">

<span id="err\_end\_data" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="salary">Salary</label>

<input type="number" class="form-control shadow-none text-custom" name="salary"

id="salary" min="0">

<span id="err\_salary" class="text text-danger text-custom"></span>

</div>

<div class="my-3" >

<label class="text-custom" for="image\_employee">Image Employee</label>

<input type="file" accept="image/jpeg" name="image" class="form-control shadow- none text-custom" id="image\_employee">

<span id="err\_image\_employee" class="text text-danger text-custom"></span>

</div>

</div>

<div class="modal-footer">

<button id="close-modal-footer" type="button" class="btn btn-danger shadow-none button\_text" data-bs-dismiss="modal">Cancel</button>

<button type="submit" class="btn btn-primary shadow-none button\_text" id="submit\_button"></button>

</div>

</form>

</div>

</div>

</div>

<div id="delete\_modal" class="modal fade">

<div class="modal-dialog">

<div class="modal-content">

<form id="delete\_form">

<input type="hidden" value="" name="delete\_id">

<div class="modal-header">

<div class="modal-title fs-2 shadow-none text-title">Delete</div>

<button class="btn btn-close shadow-none button\_text" type="button" data-bs-

dismiss="modal"></button>

</div>

<div class="modal-body">

<span class="text-custom">Are you sure you want delete

<b class="text-custom" id="delete\_name"></b>?</span>

</div>

<div class="modal-footer">

<button id="close-delete-modal-footer" class="btn btn-danger shadow-none button\_text" type="button" data-bs-dismiss="modal">Cancel</button>

<button class="btn btn-secondary shadow-none button\_text" type="submit">Delete</button>

</div>

</form>

</div>

</div>

</div>

</div>

<?php require\_once('../../../../bootstrap/script.php'); ?>

<script src="./script.js" defer></script>

</body>

</html>

design.js

*document*.addEventListener('DOMContentLoaded', (e) => {

e.preventDefault();

const searchParams = {

searchInput: *document*.getElementById('search\_inp'), searchBtn: *document*.getElementById('search\_btn')

}

*document*.getElementById('back\_btn').onclick = (e) => { e.preventDefault();

*window*.location.href = '../main/index.php';

}

searchParams.searchInput.oninput = (e) => {

e.preventDefault();

const value = e.target.value;

if(value.length > 0 && value.match(/^[a-zA-Z]+$/)){ searchParams.searchBtn.disabled = false;

}

else{

searchParams.searchBtn.disabled = true;

*document*.querySelectorAll('#table\_body tr').forEach(el => { el.classList.remove('none');

});

*document*.getElementById('empty')?.remove();

}

}

searchParams.searchBtn.onclick = async (e) => { e.preventDefault();

const val = searchParams.searchInput.value;

const formData = new FormData(); formData.append('value', val);

formData.append('action', 'search');

const response = await data(formData).then(response => response.json());

if(response.array && response.status){ checkMatch(response.array);

}

}

function checkMatch(array){

const bodyTable = *document*.getElementById('table\_body'); const rows = bodyTable.querySelectorAll('tr');

rows.forEach(row => {

const id = parseInt(row.getAttribute('data-id'));

if(array.some(el => parseInt(el.id) === id)){ row.classList.remove('none');

}

else{

row.classList.add('none');

}

});

checkLength(array);

}

function checkLength(array){ if(array.length === 0){

*document*.getElementById('table\_body').insertAdjacentHTML('beforeend', `

<tr id="empty">

<td colspan="9" class="text-title text-center" >Not Found!</td>

</tr>

`);

}

else {

*document*.getElementById('empty')?.remove();

}

}

// added functionality

const addUpdateModal = new bootstrap.Modal(*document*.getElementById('add\_update\_modal')); const deleteModal = new bootstrap.Modal(*document*.getElementById('delete\_modal'));

const bodyTable = *document*.getElementById('table\_body');

setEmpty(bodyTable);

function setEmpty(bodyTable){ if(bodyTable.querySelectorAll('tr').length === 0){

bodyTable.insertAdjacentHTML('beforeend', `

<tr id="empty">

<td colspan="9" class="text-center text-title">Table is Empty</td>

</tr>

`);

}

else{

if (*document*.getElementById('empty')){

*document*.getElementById('empty').remove();

}

}

}

const addUpdateModalParams = {

modalTitle: *document*.getElementById('title\_modal'), modalForm: *document*.getElementById('add\_update\_form'), submitForm: *document*.getElementById('submit\_button'), form: *document*.getElementById('add\_update\_form'),

closeBtnTriggerFooter: *document*.getElementById('close-modal-footer'), closeBtnTriggerHeader: *document*.getElementById('close-modal-header'), action: null

};

const deleteParams = {

form: *document*.getElementById('delete\_form'),

closeBtnTriggerFooter: *document*.getElementById('close-delete-modal-footer'),

textName: *document*.getElementById('delete\_name')

}

const alert = {

success\_message: *document*.getElementById('success\_message'), alert\_success: *document*.getElementById('alert\_success')

}

const alertAccess = {

warning\_update: *document*.getElementById('alert\_warning\_update'), warning\_add: *document*.getElementById('alert\_warning\_add'), warning\_delete: *document*.getElementById('alert\_warning\_delete')

}

const paramsDefault = {

addButton: *document*.getElementById('add'),

updateButtons: *document*.querySelectorAll('.update-btn'), deleteButtons: *document*.querySelectorAll('.delete-btn'),

};

// action on errors for update add

function modalUpdateAddErrors() { *document*.getElementById('err\_employee\_name').textContent = ''; *document*.getElementById('err\_position').textContent = ''; *document*.getElementById('err\_project\_name').textContent = ''; *document*.getElementById('err\_project\_description').textContent = ''; *document*.getElementById('err\_start\_data').textContent = ''; *document*.getElementById('err\_end\_data').textContent = ''; *document*.getElementById('err\_salary').textContent = ''; *document*.getElementById('err\_image\_employee').textContent = '';

}

function errorLog(code) { switch (code) {

case 1:

return 'err\_employee\_name';

case 2:

return 'err\_position'; case 3:

return 'err\_project\_name' case 4:

return 'err\_project\_description'; case 5:

return 'err\_start\_data';

case 6:

return 'err\_end\_data'; case 7:

return 'err\_salary'; case 8:

return 'err\_image\_employee';

}

}

// clear after close addUpdateModal.\_element.onclick = (e) => {

if(!e.target.closest('#add\_update\_form')){ modalUpdateAddErrors(); addUpdateModalParams.form.reset();

}

}

addUpdateModalParams.closeBtnTriggerHeader.onclick = (e) => { modalUpdateAddErrors();

addUpdateModalParams.form.reset();

}

addUpdateModalParams.closeBtnTriggerFooter.onclick = (e) => { modalUpdateAddErrors(); addUpdateModalParams.form.reset();

}

// update add delete actions exit paramsDefault.addButton.onclick = async (e) => {

e.preventDefault();

const formData = new FormData(); formData.append('action', 'access');

const response = await data(formData).then(response => response.json());

if(response.status){

*document*.getElementById('update\_id').value = '0';

addUpdateModalParams.opened = true;

addUpdateModalParams.modalTitle.textContent = 'Add'; addUpdateModalParams.submitForm.textContent = 'Add';

addUpdateModalParams.action = 'add';

addUpdateModal.show();

}

else{

alertAccess.warning\_add.style.setProperty('transform', 'translateY(0%)'); setTimeout(() => {

alertAccess.warning\_add.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

paramsDefault.updateButtons.forEach(btn => { buttonUpdate(btn);

});

paramsDefault.deleteButtons.forEach(btn => { buttonDelete(btn);

});

//add update submit addUpdateModalParams.form.onsubmit = async (e) => {

e.preventDefault();

const formData = new FormData(addUpdateModalParams.form);

formData.append('action', addUpdateModalParams.action);

const imageFile = *document*.getElementById('image\_employee'); formData.append('image\_employee', imageFile.files[0]);

const response = await data(formData).then(response => response.json());

modalUpdateAddErrors();

if (response.errors?.length >= 1 && response.status === false) { for (const error of response.errors) {

const className = errorLog(parseInt(error.code));

*document*.getElementById(className).textContent = error.message;

}

}

else{

if (addUpdateModalParams.action === 'add') { alert.success\_message.textContent = 'You are successfully added employee!'; add(response.data);

setEmpty(*document*.querySelector('#table\_body')); bindDataToUpdate(response.data.id); bindForDelete(response.data.id);

} else {

alert.success\_message.textContent = 'You are successfully updated employee!'; update(response.data);

}

addUpdateModalParams.closeBtnTriggerFooter.click(); modalUpdateAddErrors(); addUpdateModalParams.form.reset();

alert.alert\_success.style.setProperty('transform', 'translateY(0%)'); setTimeout(() => {

alert.alert\_success.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

// delete submit

deleteParams.form.onsubmit = async (e) => { e.preventDefault();

const formDataDelete = new FormData(deleteParams.form);

formDataDelete.append('action', 'delete');

const response = await data(formDataDelete).then(response => response.json());

if (response.status) { deleteParams.closeBtnTriggerFooter.click();

*document*.querySelector(`#table\_body tr[data-id="${parseInt(response.delete\_id)}"]`).remove(); setEmpty(bodyTable);

alert.success\_message.textContent = 'You are successfully deleted employee!';

alert.warning\_add.style.setProperty('transform', 'translateY(0%)'); setTimeout(() => {

alert.alert\_success.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

// html add, update function add(data) {

const newData = `

<tr data-id="${data.id}">

<td class="employee\_name" >${data.employee\_name}</td>

<td class="position" >${data.position}</td>

<td class="project\_name" >${data.project\_name}</td>

<td class="project\_description" >${data.project\_description}</td>

<td class="start\_date" >${data.start\_date}</td>

<td class="end\_date" >${data.end\_date}</td>

<td class="salary" >${data.salary}</td>

<td class="image\_employee">

<img src="./dd/DD${data.id}.jpg" style="width: 75px; height: 75px; object-fit: cover;" alt="Emploee Image">

</td>

<td>

<div class="d-flex align-items-center flex-row">

<button data-delete="${data.id}" type="button" class="border border-end-0 rounded rounded-end-0 delete-btn">

<i class='fa fa-trash'></i>

</button>

<button data-update="${data.id}" type="button" class="border rounded rounded-start-0 update-btn">

<i class="fa fa-pen"></i>

</button>

</div>

</td>

</tr>

`;

bodyTable.insertAdjacentHTML('beforeend', newData);

}

function update(data) {

const id\_update = parseInt(data.id);

const updateHtml = *document*.querySelector(`#table\_body tr[data-id="${id\_update}"]`);

if (updateHtml) {

updateHtml.querySelector('.employee\_name').textContent = data.employee\_name; updateHtml.querySelector('.position').textContent = data.position; updateHtml.querySelector('.project\_name').textContent = data.project\_name; updateHtml.querySelector('.project\_description').textContent = data.project\_description; updateHtml.querySelector('.start\_date').textContent = data.start\_date; updateHtml.querySelector('.end\_date').textContent = data.end\_date; updateHtml.querySelector('.salary').textContent = data.salary;

const imgElement = *document*.querySelector(`img[data-img="${id\_update}"]`); if (imgElement) {

const timestamp = new *Date*().getTime();

imgElement.src = `./dd/DD${id\_update}.jpg?timestamp=${timestamp}`;

}

}

}

// bind after add (update, delete) function bindDataToUpdate(id) {

const btn = *document*.querySelector(`#table\_body tr[data-id="${parseInt(id)}"] td button[data- update="${parseInt(id)}"]`);

if (btn) { buttonUpdate(btn);

}

}

function buttonUpdate(btn) { if (btn) {

btn.onclick = async (e) => { e.preventDefault();

const formData = new FormData(); formData.append('action', 'access');

const response = await data(formData).then(response => response.json());

if(response.status){ addUpdateModalParams.modalTitle.textContent = 'Update'; addUpdateModalParams.submitForm.textContent = 'Update';

addUpdateModalParams.action = 'update';

const getId = parseInt(btn.getAttribute('data-update'));

const getData = *document*.querySelector(`#table\_body tr[data-id="${getId}"]`);

if(getData){ addUpdateModalParams.form.elements['update\_id'].value = getId;

addUpdateModalParams.form.elements['employee\_name'].value = getData.querySelector('.employee\_name').textContent;

addUpdateModalParams.form.elements['position'].value = getData.querySelector('.position').textContent;

addUpdateModalParams.form.elements['project\_name'].value =

getData.querySelector('.project\_name').textContent;

addUpdateModalParams.form.elements['project\_description'].value = getData.querySelector('.project\_description').textContent;

addUpdateModalParams.form.elements['start\_date'].value = getData.querySelector('.start\_date').textContent;

addUpdateModalParams.form.elements['end\_date'].value = getData.querySelector('.end\_date').textContent;

addUpdateModalParams.form.elements['salary'].value = getData.querySelector('.salary').textContent;

addUpdateModal.show();

}

}

else{

alertAccess.warning\_update.style.setProperty('transform', 'translateY(0%)');

setTimeout(() => {

alertAccess.warning\_update.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

}

}

// delete bind

function bindForDelete(id) {

const btn = document.querySelector(`#table\_body tr[data-id="${id}"] td button[data- delete="${id}"]`);

if (btn) { buttonDelete(btn);

}

}

function buttonDelete(btn) { if (btn) {

btn.onclick = async (e) => { e.preventDefault();

const formData = new FormData(); formData.append('action', 'access');

const response = await data(formData).then(response => response.json());

if(response.status) {

const getId = parseInt(btn.getAttribute('data-delete'));

const getData = document.querySelector(`#table\_body tr[data-id="${getId}"]`);

if (getData) {

deleteParams.form.elements['delete\_id'].value = getId; document.getElementById('delete\_name').textContent =

getData.querySelector('.employee\_name').textContent; deleteModal.show();

}

}

else{

alertAccess.warning\_delete.style.setProperty('transform', 'translateY(0%)');

setTimeout(() => {

alertAccess.warning\_delete.style.setProperty('transform', 'translateY(150%)');

}, 2000);

}

}

}

}

async function data(data){

const url = "http://localhost:63342/furniture\_factory/view/mainPage/workStaff/design/validate.php";

}

});

return await fetch(url, { method: 'POST', body: data

});

(Design) validate.php

<?php

header('Access-Control-Allow-Origin: http://localhost:63342/\*'); header('Access-Control-Allow-Methods: POST');

require\_once('../../../../sqlMain.php');

$sqlMain = new MainSql();

if($\_SERVER['REQUEST\_METHOD'] === 'POST'){

$request = $\_POST;

if(isset($request['action'])){

if($request['action'] === 'access'){

$username = $sqlMain->getUsername();

if($sqlMain->checkUsernameAdmin($username)){

echo json\_encode(['status' => true]);

}

else{

echo json\_encode(['status' => false]);

}

die();

}

if($request['action'] === 'search'){

$value = trim($request['value']);

if($arrayIds = $sqlMain->search($value, "design\_development")){

echo json\_encode(['status' => true, 'array' => $arrayIds]);

}

else{

echo json\_encode(['status' => true, 'array' => $arrayIds]);

}

die();

}

if($request['action'] === 'add' || $request['action'] === 'update'){

$errors = [];

if(empty($request['employee\_name'])){

$errors[] = ['code' => 1, 'message' => "You don't fill employee name field!"];

}

if(empty($request['position'])){

$errors[] = ['code' => 2, 'message' => "You don't fill position field!"];

}

if(empty($request['project\_name'])){

$errors[] = ['code' => 3, 'message' => "You don't fill project name field!"];

}

if(empty(htmlspecialchars($request['project\_description']))){

$errors[] = ['code' => 4, 'message' => "You don't fill project description field!"];

}

if(empty($request['start\_date'])){

$errors[] = ['code' => 5, 'message' => "You don't fill start date field!"];

}

if(empty($request['end\_date'])){

$errors[] = ['code' => 6, 'message' => "You don't fill end date field!"];

}

if(empty($request['salary'])){

$errors[] = ['code' => 7, 'message' => "You don't fill salary field!"];

}

if(empty($\_FILES['image\_employee']) && $request['action'] === 'add'){

$errors[] = ['code' => 8, 'message' => "You don't upload employee image!"];

}

if(!empty($errors)){

echo json\_encode(['status' => false, 'errors' => $errors]); die();

}

else{

if($request['action'] === 'add'){

if($data = $sqlMain->addDesign($request, 'design\_development')){

$idAdded = (int)$data['id'];

$tmpName = $\_FILES['image\_employee']['tmp\_name'];

$path = "./dd/DD" . $idAdded . ".jpg";

move\_uploaded\_file($tmpName, $path);

echo json\_encode(['status' => true, 'data' => $data]);

}

}

if($request['action'] === 'update'){

if($data = $sqlMain->updateDesign($request, 'design\_development')){

if(isset($\_FILES['image\_employee'])){

$idUpdated = (int)$data['id'];

$tmpName = $\_FILES['image\_employee']['tmp\_name']; if($tmpName){

$path = "./dd/DD" . $idUpdated . ".jpg";

move\_uploaded\_file($tmpName, $path);

}

}

echo json\_encode(['status' => true, 'data' => $data]);

}

}

die();

}

}

if($request['action'] === 'delete'){

if($deleteRes = $sqlMain->delete($request['delete\_id'], 'design\_development')){

$id = (int)$request['delete\_id'];

$path = "./dd/DD" . $id . ".jpg"; if(file\_exists($path)){

unlink($path);

}

echo json\_encode(['status' => true, 'delete\_id' => $id]);

die();

}

}

}

}