

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Керівник роботи

_____ Ілля АХАЛАДЗЕ

“ ____ ” _____ 2024 р.

ТЕМА КУРСОВОЇ РОБОТИ

Текст програми

КПІ.ПІ-1313.045440.06.12

“ПОГОДЖЕНО”

Керівник роботи:

_____ Ілля АХАЛАДЗЕ

Консультант:

Виконавець:

_____ Дмитро ЗАМКОВИЙ

Київ – 2024

Файл index.php

<?

/**

* License

*

* Copyright (c) 2023 Dmytro Zamkovyi

*

* You are prohibited from:

*

* - Using any part of this code for any purpose.

* - Copying or reproducing any part of this code in any form.

* - Modifying or altering any part of this code.

*

* See the full text of the license in the LICENSE file in the
root directory of this repository.

*/

declare(strict_types=1);

require_once __DIR__.'./sys.php';

Curswork\ActionClass::start();

Файл config.php

<?

\$dbhost = '';

\$dbname = '';

\$dbuser = '';

\$dbpswd = '';

\$dbport = '';

\$timeZone = 'Europe/Kyiv';

\$sessionKey = 'session_key';

\$salt = '';

\$projectName = 'Курсова робота';

Файл sys.php

<?

```
/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

namespace Curswork;

/**
 * Class ActionClassModel
 *
 * Клас для роботи з моделлю
 *
 * @package Curswork
 * @property ConfigClass $config - об'єкт з налаштуваннями
 * @property DBClass $db - об'єкт для роботи з базою даних
 * @property SessionClass $session - об'єкт для роботи з сесією
 * @property AuthClass $auth - об'єкт для роботи з авторизацією
 * @property RequestClass $request - об'єкт з даними запиту
 * @property ResponseClass $response - об'єкт для роботи з
Відповіддю
 */
```

```

class ActionClassModel
{
    public ?AuthClass $auth = null;
    public ?DBClass $db = null;
    public ?SessionClass $session = null;
    public ?ConfigClass $config = null;
    public ?RequestClass $request = null;
    public ?ResponseClass $response = null;
}

/**
 * Class ActionClass
 *
 * Основний клас для роботи з системою
 *
 * @package Curswork
 * @property ConfigClass $config - об'єкт з налаштуваннями
 * @property DBClass $db - об'єкт для роботи з базою даних
 * @property RequestClass $request - об'єкт з даними запиту
 * @property SessionClass $session - об'єкт для роботи з сесією
 * @property AuthClass $auth - об'єкт для роботи з авторизацією
 * @property ResponseClass $response - об'єкт для роботи з
Відповіддю
 * @property ViewClass $view - об'єкт для роботи з відображенням
 * @property string $module - назва модуля, перший рівень шляху
 * @property string $class - назва класу, другий рівень шляху
 * @property string $action - назва методу, третій рівень шляху
 * @property string|null $params - параметри, четвертий рівень
шляху
 * @property string $dir - шлях до папки з поточним класом
 * @property object|null $model - об'єкт моделі
 * @method string load(string $module, string $class, string
$action, ?string $params = '', array &$data = array(), bool
$template = true) - функція для підключення контролера
 * @method void load_model(string $module, string $class) -
функція для підключення моделі

```

```

* @method string extract_route(string $route) - функція для
розбиття шляху на модуль, клас, метод та параметри
* @method string start_route(string $module, string $class,
string $action, ?string $params, array &$data = array(), bool
$internal = false) - функція для запуску модуля, класу, методу з
параметрами
* @method void start() - точка входу в систему
* @method void __construct(string $module, string $class,
string $action, ?string $params, string $dir) - конструктор
*/
class ActionClass
{
    private static ?ConfigClass $_config = null;
    private static ?DBClass $_db = null;
    private static ?RequestClass $_request = null;
    private static ?SessionClass $_session = null;
    private static ?AuthClass $_auth = null;
    private static ?ResponseClass $_response = null;

    public ?ConfigClass $config = null;
    public ?DBClass $db = null;
    public ?RequestClass $request = null;
    public ?SessionClass $session = null;
    public ?AuthClass $auth = null;
    public ?ResponseClass $response = null;
    public ?ViewClass $view = null;

    public string $module = '';
    public string $class = '';
    public string $action = '';
    public ?string $params = null;

    public string $dir = '';
    public ?object $model = null;

    /**

```

```

* ActionController constructor.
*
* @param string $module - назва модуля
* @param string $class - назва класу
* @param string $action - назва методу
* @param string|null $params - параметри
* @param string $dir - шлях до папки з поточним класом
*/
public function __construct(string $module, string $class,
string $action, ?string $params, string $dir)
{
    $this->module = $module;
    $this->class = $class;
    $this->action = $action;
    $this->params = $params;
    $this->dir = $dir;

    $this->config = ActionController::$_config;
    $this->db = ActionController::$_db;

    if (ActionController::$_request === null) {
        ActionController::$_request = new RequestClass($this);
    }
    $this->request = ActionController::$_request;

    if (ActionController::$_session === null) {
        ActionController::$_session = new SessionClass($this);
    }
    $this->session = ActionController::$_session;

    if (ActionController::$_auth === null) {
        ActionController::$_auth = new AuthClass($this);
    }
    $this->auth = ActionController::$_auth;

    if (ActionController::$_response === null) {

```

```

        ActionClass::$_response = new ResponseClass($this);
    }
    $this->response = ActionClass::$_response;

    $this->view = new ViewClass($this);
}

/**
 * Точка входа в систему
 *
 * @return void
 */
public static function start()
{
    if (ActionClass::$_config === null) {
        ActionClass::$_config = new
ConfigClass('/config.php');
    }

    if (ActionClass::$_db === null) {
        ActionClass::$_db = new
DBClass(ActionClass::$_config);
    }

    if (isset($_SERVER['REQUEST_URI'])) {
        $uri = explode('?', $_SERVER['REQUEST_URI'])[0];
        $uri = rtrim($uri, '/');
        $uri = urldecode($uri);
        list($module, $class, $action, $params) =
ActionClass::extract_route($uri);
    } else {
        exit('Variable $_SERVER[\'REQUEST_URI\'] not
found');
    }

    echo ActionClass::start_route($module, $class, $action,
$params) ?? '';
}

```

```

    }

    /**
     * Функція для розбиття шляху на модуль, клас, метод та
    параметри
     *
     * @param string $route - шлях
     * @return array - масив з модулем, класом, методом та
    параметрами
     */
    public static function extract_route(string $route)
    {
        $load = explode('/', $route, 5);
        if (isset($load[1]) && strlen($load[1]) > 0) {
            $module = $load[1];
        } else {
            $module = 'index';
        }
        if (isset($load[2]) && strlen($load[2]) > 0) {
            $class = $load[2];
        } else {
            $class = 'index';
        }
        if (isset($load[3]) && strlen($load[3]) > 0) {
            $action = $load[3];
        } else {
            $action = 'index';
        }

        if (isset($load[4]) && strlen($load[4]) > 0) {
            $params = $load[4];
        } else {
            $params = '';
        }
        return [$module, $class, $action, $params];
    }

```



```

/**
 * Функція для запуску модуля, класу, методу з параметрами
 *
 * @param string $module - назва модуля
 * @param string $class - назва класу
 * @param string $action - назва методу
 * @param string|null $params - параметри
 * @param array $data - дані для виводу
 * @param bool $internal - чи внутрішній запит
 * @return string - результат виконання
 */
public static function start_route(string $module, string
$class, string $action, ?string $params, array &$data = array(),
bool $internal = false)
{
    if (!(bool)preg_match('/^[a-z0-9_]+$/', $module) ||
!(bool)preg_match('/^[a-z0-9_]+$/', $class) ||
!(bool)preg_match('/^[a-z0-9_]+$/', $action)) {
        exit;
    }

    $dir = __DIR__ . '/app/' . $module . '/' . $class;
    $file = $dir . '/action.php';
    $is_404 = false;
    $is_login_error = false;

    if (file_exists($file)) {
        require_once $file;
        $className = $module . ucfirst($class);
        if (class_exists($className)) {
            $className = new $className($module, $class,
$action, $params, $dir);
            if ($className->auth->get_access() || $internal)
{
                $actionName = 'action_' . $action;

```

```

        if (method_exists($className, $actionName))
    {

        $className->load_model($module, $class);
        return $className->$actionName($params,
$data);

        } else {
            $is_404 = true;
        }
    } else {
        $is_login_error = true;
    }
} else {
    $is_404 = true;
}

} else {
    $is_404 = true;
}

if ($is_404) {
    header('HTTP/1.1 404 Not Found');
    $dir = __DIR__ . '/app/index/index';
    $file = $dir . '/action.php';
    require_once $file;
    $className = new \indexIndex('index', 'index',
'404', '', $dir);
    return $className->action_404('', $data);
}

if ($is_login_error) {
    if (isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
$_SERVER['HTTP_X_REQUESTED_WITH'] == 'XMLHttpRequest') {
        $className = new \indexIndex($module, $class,
$action, $params, $dir);
        $className->load_model($module, $class);
        $className->auth();
        return '';
    }
}

```

```

    }
    header('HTTP/1.1 401 Unauthorized');
    $dir = __DIR__ . '/app/index/index';
    $file = $dir . '/action.php';
    require_once $file;
    $className = new \indexIndex('index', 'index',
'login', '', $dir);
    return $className->action_login('', $data);
}
return '';
}

/**
 * Функція для підключення моделі
 *
 * @param string $module - назва модуля
 * @param string $class - назва класу
 * @return void
 */
public function load_model(string $module, string $class)
{
    $class_name = get_class($this);
    $file = $this->dir . '/model.php';
    if (file_exists($file)) {
        require_once $file;
        $class_name .= 'Model';
        if (class_exists($class_name)) {
            $this->model = new $class_name($module, $class);
        } else {
            $this->model = new \stdClass();
        }
    } else {
        $this->model = new \stdClass();
    }

    $this->model->auth = $this->auth;

```

```

        $this->model->db = $this->db;
        $this->model->session = $this->session;
        $this->model->config = $this->config;
        $this->model->request = $this->request;
        $this->model->response = $this->response;
    }

    /**
     * Функція для підключення контролера
     *
     * @param string $module - назва модуля
     * @param string $class - назва класу
     * @param string $action - назва методу
     * @param string|null $params - параметри
     * @param array $data - дані для виводу
     * @param bool $template - чи виводити шаблон
     * @return string - результат виконання
     */
    public function load(string $module, string $class, string
$action, ?string $params = '', array &$data = array(), bool
$template = true)
    {
        if ($template) {
            ob_start();
        }
        $res = ActionController::start_route($module, $class,
$action, $params, $data, true);
        if ($template) {
            return ob_get_clean();
        }
        return $res;
    }
}

/**

```

```

* Class ConfigClass
*
* Клас для роботи з налаштуваннями
*
* @package Curswork
* @method void __construct(string $filename) - конструктор
* @method string __get(string $name) - функція для отримання
значення елементу
* @method bool __isset(string $name) - функція для перевірки
наявності елементу
*/
class ConfigClass
{
    private array $_config = [];

    /**
     * ConfigClass constructor.
     *
     * @param string $filename - шлях до файлу з налаштуваннями
     */
    public function __construct($filename)
    {
        // Якщо php не зміг підключити файл - виникає помилка
        require_once __DIR__ . $filename;
        $this->_config = get_defined_vars();
        unset($this->_config['filename']);

        if (isset($this->timeZone)) {
            date_default_timezone_set($this->timeZone);
        }

        if (isset($this->timeExecution)) {
            set_time_limit($this->timeExecution);
        }
    }
}

```

```

/**
 * Функція для отримання значення елементу
 *
 * @param string $name - ключ елементу
 * @return mixed|null - значення елементу
 */
public function __get(string $name)
{
    return $this->_config[$name];
}

/**
 * Функція для перевірки наявності елементу
 *
 * @param string $name - ключ елементу
 * @return bool - чи існує елемент
 */
public function __isset(string $name)
{
    return isset($this->_config[$name]);
}
}

/**
 * Class DBClass
 *
 * Клас для роботи з базою даних
 *
 * @package Curswork
 * @property string $last_sql - останній запит
 * @property bool $last_error - чи була помилка
 * @property string $last_error_string - остання помилка
 * @property int $count_rows - кількість рядків
 * @method void __construct(ConfigClass $config) - конструктор
 * @method void __destruct() - деструктор

```

```

    * @method array|null query(string $query) - функція для
виконання запиту до бази даних
    * @method string escape(string $data) - функція для екранування
даних
    */
class DBClass
{
    private $_db = null;

    private string $_dbhost;
    private string $_dbname;
    private string $_dbuser;
    private string $_dbpswd;
    private string $_dbport;

    public string $last_sql = '';
    public bool $last_error = false;
    public string $last_error_string = '';
    public int $count_rows = 0;

    /**
     * DBClass constructor.
     *
     * @param ConfigClass $config - об'єкт з налаштуваннями
     */
    public function __construct(ConfigClass &$config)
    {
        if (isset($config->dbhost)) {
            $this->_dbhost = $config->dbhost;
        } else {
            $this->_dbhost = 'localhost';
        }

        if (isset($config->dbname)) {
            $this->_dbname = $config->dbname;
        } else {

```

```

        exit('Не вказано назву бази даних');
    }

    if (isset($config->dbuser)) {
        $this->_dbuser = $config->dbuser;
    } else {
        $this->_dbuser = 'postgres';
    }

    if (isset($config->dbpswd)) {
        $this->_dbpswd = $config->dbpswd;
    } else {
        exit('Не вказано пароль користувача бази даних');
    }

    if (isset($config->dbport)) {
        $this->_dbport = $config->dbport;
    } else {
        $this->_dbport = '5432';
    }
}

/**
 * DBClass destructor.
 *
 * @return void
 */
public function __destruct()
{
    if ($this->_db !== null) {
        register_shutdown_function(function ($db) {
            pg_close($db);
        }, $this->_db);
    }
}

```



```

/**
 * Функція для підключення до бази даних
 *
 * @return void
 * @throws \RuntimeException - помилка підключення до бази
даних
 */
private function connect()
{
    $connect_string = 'host=\' ' . addslashes($this->_dbhost)
. '\ ' ';
    $connect_string .= 'port=\' ' . addslashes($this-
>_dbport) . '\ ' ';
    $connect_string .= 'user=\' ' . addslashes($this-
>_dbuser) . '\ ' ';
    $connect_string .= 'dbname=\' ' . addslashes($this-
>_dbname) . '\ ' ';
    $connect_string .= 'password=\' ' . addslashes($this-
>_dbpswd) . '\ ' ';
    $connect_string .= 'options=\'--client_encoding=UTF8\'
';

    $this->_db = pg_connect($connect_string);
    if ($this->_db === false) {
        throw new \RuntimeException('Unable to connect to
db');
    }
}

/**
 * Функція для виконання запиту до бази даних
 *
 * @param string $query - запит
 * @return array|null - результат запиту
 */
public function query(string $query)
{

```

```

        if ($this->_db == null) {
            $this->connect();
        }
        $this->last_sql = $query;
        $res = @pg_query($this->_db, $query);
        if ($res === false) {
            $this->last_error = true;
            $this->last_error_string = pg_last_error($this-
>_db);

            return null;
        }
        $this->last_error = false;
        $this->last_error_string = '';
        $count_rows = pg_num_rows($res);
        if ($count_rows == 0) {
            $result = array();
        } else {
            $result = pg_fetch_all($res);
            if ($result === false) {
                $result = array();
            }
        }
        pg_free_result($res);
        return $result;
    }

/**
 * Функція для екранування даних
 *
 * @param string $data - дані
 * @return string - екрановані дані
 */
public function escape(string $data)
{
    if ($this->_db === null) {
        $this->connect();
    }

```

```

        }
        return pg_escape_string($this->_db, $data);
    }
}

/**
 * Class AuthClass
 *
 * Клас для роботи з авторизацією
 *
 * @package Curswork
 * @method void __construct(ActionClass $action) - конструктор
 * @method bool get_access() - функція для перевірки доступу
 */
class AuthClass
{
    private ActionClass $_action;

    /**
     * AuthClass constructor.
     *
     * @param ActionClass $action
     */
    public function __construct(ActionClass &$action)
    {
        $this->_action = $action;
    }

    /**
     * Функція для перевірки доступу
     *
     * @return void
     */
    public function get_access()
    {

```

```

        if ($this->_action->session->system) {
            return true;
        }
        if ($this->_action->session->user_id != null) {
            $res = $this->_action->db->query('SELECT
role.role_name as role FROM employee, role WHERE id_role =
role.id AND employee.id = ' . $this->_action->session->user_id .
';');

            if ($res) {
                if ($res[0]['role'] == $this->_action->module ||
$this->_action->module == 'index') {
                    return true;
                }
                return false;
            }
            return false;
        }
        return false;
    }
}

```

```

/**
 * Class SessionClass
 *
 * Клас для роботи з сесією
 *
 * @package Curswork
 * @property string $session - ключ сесії
 * @property int|null $user_id - ідентифікатор користувача
 * @method void __construct(ActionClass $action) - конструктор
 * @method array load() - функція для завантаження сесії
 * @method void save() - функція для збереження сесії
 * @method void __set(string $key, $value) - функція для
встановлення значення елементу

```

```

    * @method mixed|null __get(string $key) - функція для отримання
значення елементу
    * @method bool __isset(string $key) - функція для перевірки
наявності елементу
    * @method void __unset(string $key) - функція для видалення
елементу
    * @method void __destruct() - деструктор
    */
class SessionClass
{
    public const ON_YEAR = 60 * 60 * 24 * 365;

    private ?ActionClass $_action = null;
    private string $_session = '';
    private ?int $_session_id = null;
    private array $_data = array();
    private ?int $_user_id = null;
    private array $_changed = array();

    /**
     * SessionClass constructor.
     *
     * @package Curswork
     * @param ActionClass $action
     */
    function __construct(ActionClass &$action)
    {
        $this->_action = $action;
        $key = $action->config->sessionKey;
        $session = $action->request->cookie->{$key} ?? '';
        if (!(bool)preg_match('/^[a-z0-9]{128}$/', $session)) {
            $session = hash('sha512', $this->_action->config-
>salt . $action->request->ip . microtime());
            $action->request->cookie->{$key} = $session;
            setcookie($key, $session, time() +
SessionClass::ON_YEAR, '/');

```

```

    }

    $this->_session = $session;
    $this->_data = $this->load();
}

/**
 * SessionClass destructor.
 *
 * @return void
 */
public function __destruct()
{
    $this->save();
}

/**
 * Функція для завантаження сесії
 *
 * @return array - дані сесії
 */
public function load()
{
    $sql = 'SELECT * FROM "session" WHERE session_key = \''
. $this->_session . '\';';
    $query = (array)$this->_action->db->query($sql);
    if (count($query) > 0) {
        $this->_session_id = (int)$query[0]['id'];
        $this->_user_id = $query[0]['id_employee'] === null
? null : (int)$query[0]['id_employee'];
        $data = unserialize($query[0]['data']);
    } else {
        $sql = 'INSERT INTO "session" (session_key,
session_create_date, session_last_date, "data") VALUES (\''
. $this->_session . '\', now(), now(), \'' . serialize(array()) .
'\') RETURNING id;';
    }
}

```

```

        $query = (array)$this->_action->db->query($sql);
        $this->_session_id = (int)$query[0]['id'];
        $this->_user_id = null;
        $data = array();
    }
    return $data;
}

/**
 * Функція для збереження сесії
 *
 * @return void
 */
public function save()
{
    if (count($this->_changed) > 0) {
        $user_id = (string)$this->_user_id;
        $data = $this->_data;
        unset($data['user_id']);
        $json = $this->_action->db->escape(serialize($data));
        $sql = 'UPDATE "session" SET "data" = \'' . $json .
        '\', session_last_date = now() WHERE session_key = \'' . $this->_session .
        '\';';
        $this->_action->db->query($sql);
    } else {
        $sql = 'UPDATE "session" SET session_last_date =
        now() WHERE session_key = \'' . $this->_session . '\';';
        $this->_action->db->query($sql);
    }
}

/**
 * Функція для встановлення значення елементу
 *
 * @param string $key - ключ елементу

```

```

    * @param mixed $value - значення елементу
    */
public function __set(string $key, $value)
{
    if (!in_array($key, $this->_changed)) {
        $this->_changed[] = $key;
    }
    $this->_data[$key] = $value;
}

/**
 * Функція для отримання значення елементу
 *
 * @param string $key - ключ елементу
 * @return mixed|null - значення елементу
 */
public function __get(string $key)
{
    if (isset($this->_data[$key])) {
        return $this->_data[$key];
    }

    if ($key == 'user_id') {
        return $this->_user_id;
    }

    if ($key == 'session') {
        return $this->_session;
    }

    return null;
}

/**
 * Функція для перевірки наявності елементу
 *
```



```

    * @param string $key - ключ елементу
    * @return bool - чи існує елемент
    */
    public function __isset(string $key)
    {
        return isset($this->_data[$key]);
    }

    /**
     * Функція для видалення елементу
     *
     * Функція встановлює значення елементу в null.
     *
     * @param string $key - ключ елементу
     * @return void
     */
    public function __unset(string $key)
    {
        if (!in_array($key, $this->_changed)) {
            $this->_changed[] = $key;
        }
        unset($this->_data[$key]);
    }
}

/**
 * Class ParameterClass
 *
 * Клас для роботи з параметрами
 *
 * @package Curswork
 * @property int $count - кількість елементів
 * @method void __construct(array $data) - конструктор
 * @method void __set(string $key, $value) - функція для
встановлення значення елементу

```

```

    * @method mixed|null __get(string $key) - функція для отримання
значення елементу
    * @method bool __isset(string $key) - функція для перевірки
наявності елементу
    * @method void __unset(string $key) - функція для видалення
елементу
    */
class ParameterClass
{
    private array $_data = array();
    private int $_count = 0;

    /**
     * ParameterClass constructor.
     *
     * @param array $data - дані
     */
    public function __construct(array $data)
    {
        foreach ($data as $key => $value) {
            $this->_data[$key] = $value;
            $this->_count++;
        }
    }

    /**
     * Функція для отримання кількості елементів
     *
     * @return int - кількість елементів
     */
    public function count()
    {
        return $this->_count;
    }

    /**

```

```

* Функція для встановлення значення елементу
*
* @param string $key - ключ елементу
* @param mixed $value - значення елементу
*/
public function __set(string $key, $value)
{
    if (isset($this->_data[$key])) {
        $this->_data[$key] = $value;
    } else {
        $this->_data[$key] = $value;
        $this->_count++;
    }
}

/**
* Функція для отримання значення елементу
*
* @param string $key - ключ елементу
* @return mixed|null - значення елементу
*/
public function __get(string $key)
{
    return $this->_data[$key] ?? null;
}

/**
* Функція для перевірки наявності елементу
*
* @param string $key - ключ елементу
* @return bool - чи існує елемент
*/
public function __isset(string $key)
{
    return isset($this->_data[$key]);
}

```

```

/**
 * Функція для видалення елементу
 *
 * Функція встановлює значення елементу в null.
 *
 * @param string $key - ключ елементу
 */
public function __unset(string $key)
{
    if (isset($this->_data[$key])) {
        $this->_data[$key] = null;
        $this->_count--;
    }
}
}

/**
 * Class RequestClass
 *
 * Клас для роботи з даними запиту
 *
 * @package Curswork
 * @property ParameterClass $get - параметри GET
 * @property ParameterClass $post - параметри POST
 * @property ParameterClass $cookie - параметри COOKIE
 * @property string $ip - IP-адреса
 * @method void __construct(ActionClass $action) - конструктор
 */
class RequestClass
{
    private ?ActionClass $_action = null;

    public ParameterClass $get;
    public ParameterClass $post;

```

```

public ParameterClass $cookie;

public string $ip = '';

/**
 * RequestClass constructor.
 *
 * @param ActionClass $action
 */
public function __construct(ActionClass &$action)
{
    $this->_action = $action;

    $this->cookie = new ParameterClass($_COOKIE);
    $this->get = new ParameterClass($_GET);
    $this->post = new ParameterClass($_POST);

    if (isset($_SERVER['REMOTE_ADDR'])) {
        $this->ip = $_SERVER['REMOTE_ADDR'];
    }
}

/**
 * Class ResponseClass
 *
 * Клас для роботи з відповіддю
 *
 * @package Curswork
 * @property array $header - заголовки
 * @method void __construct(ActionClass $action) - конструктор
 * @method void redirect(string $url, int $status = 302) -
функція для перенаправлення на сторінку
 */
class ResponseClass

```

```

{
    private ?ActionClass $_action = null;
    public array $header = array();

    /**
     * ResponseClass constructor.
     *
     * @param ActionClass $action
     */
    public function __construct(ActionClass &$action)
    {
        $this->_action = $action;
    }

    /**
     * Функція для перенаправлення на сторінку
     *
     * @param string $header - заголовок
     * @return void
     */
    public function redirect(string $url, int $status = 302)
    {
        header('Location: ' . $url, true, $status);
        exit();
    }
}

/**
 * Class ViewClass
 *
 * Клас для роботи з відображенням
 *
 * @package Curswork
 * @method void __construct(ActionClass $action) - конструктор

```

```

    * @method void out(string|null $view = null, array $data =
array()) - функція для виводу з хедерами
    * @method string render(string $view, array $data = array()) -
функція для виводу для рендерингу в'юшки
    */
class ViewClass
{
    private ?ActionClass $_action = null;

    /**
     * ViewClass constructor.
     *
     * @param ActionClass $action
     */
    public function __construct(ActionClass &$action)
    {
        $this->_action = $action;
    }

    /**
     * Функція для виводу з хедерами
     *
     * @param string|null $view - шаблон
     * @param array $data - дані для виводу
     * @return void
     */
    public function out(?string $view = null, array &$data =
array())
    {
        foreach ($this->_action->response->header as $head) {
            header($head, false);
        }
        if (empty($view)) {
            return;
        }
        echo $this->render($view, $data);
    }
}

```

```

    }

    /**
     * Функція для виводу для рендерингу в'юшки
     *
     * @param string|null $view - шаблон
     * @param array $data - дані для виводу
     * @return void
     */
    public function render(string $view, array &$data = array())
    {
        $file = $this->_action->dir . '/view_' . $view . '.php';
        if (file_exists($file)) {
            extract($data);
            ob_start();
            require $file;
            return ob_get_clean();
        }
        return '';
    }
}

```

Файл app/admin/client/action.php

<?

```

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.

```



```

* - Modifying or altering any part of this code.
*
* See the full text of the license in the LICENSE file in the
root directory of this repository.
*/

```

```

class adminClient extends Curswork\ActionClass
{
    /**
     * Контроллер для відображення списку клієнтів
     *
     * @param $params - параметри
     * @param $data - дані
     * @return mixed
     */
    public function action_index($params, $data)
    {
        $view = "index";
        $data['label'] = 'Перегляд клієнтів';
        $data['menu'] = array(
            array(
                'name' => 'Працівники',
                'href' => '/admin/employee/index',
                'active' => false,
                'img' => '/data/icon/employee.png'
            ),
            array(
                'name' => 'Послуги',
                'href' => '/admin/service/index',
                'active' => false,
                'img' => '/data/icon/service.png'
            ),
            array(
                'name' => 'Замовлення',
                'href' => '/admin/order/index',
                'active' => false,

```

```

        'img' => '/data/icon/order.png'
    ),
    array(
        'name' => 'Клієнти',
        'href' => '/admin/client/index',
        'active' => true,
        'img' => '/data/icon/client_activ.png'
    ),
    array(
        'name' => 'Вихід',
        'href' => '/index/index/logout',
        'active' => false,
        'img' => '/data/icon/logout.png'
    ),
);
$data['head'] = $this->load('index', 'index', 'head',
'', $data);
$data['title'] = 'Клієнти · ' . $this->config-
>projectName;

$this->view->out($view, $data);
}

/**
 * Контроллер для додавання нового клієнта
 *
 * @param $params - параметри
 * @param $data - дані
 * @return mixed
 */
public function action_create($params, $data)
{
    $data['first_name'] = urldecode($this->request->post-
>first_name);
    $data['last_name'] = urldecode($this->request->post-
>last_name);

```

```

        $data['father_name'] = urldecode($this->request->post->father_name);
        $data['phone_number'] = urldecode($this->request->post->phone_number);
        $data['discount'] = (int)urldecode($this->request->post->discount);

        preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        $matches = array_slice($matches, 1);
        for ($i = 0; $i < count($matches); $i++) {
            $matches[$i] = (int)$matches[$i][0];
        }
        $data['phone_number'] = (int)implode($matches);

        if ($data['discount'] < 0 || $data['discount'] > 100) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($this->model->create($data)) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }

    /**
     * Контроллер для пошуку клієнтів
     *
     * @param $params - параметри

```

```

* @param $data - дані
* @return mixed
*/
public function action_search($params, $data)
{
    $data['first_name'] = $this->request->post->first_name;
    $data['last_name'] = $this->request->post->last_name;
    $data['father_name'] = $this->request->post-
>father_name;
    $data['phone_number'] = $this->request->post-
>phone_number;

    if ($data['first_name'] == ''){
        unset($data['first_name']);
    } else {
        $data['first_name'] =
urldecode($data['first_name']);
    }

    if ($data['last_name'] == ''){
        unset($data['last_name']);
    } else {
        $data['last_name'] = urldecode($data['last_name']);
    }

    if ($data['father_name'] == ''){
        unset($data['father_name']);
    } else {
        $data['father_name'] =
urldecode($data['father_name']);
    }

    if ($data['phone_number'] == ''){
        unset($data['phone_number']);
    } else {

```

```

        $data['phone_number'] =
urldecode($data['phone_number']);
        preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-
9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-
9]{2})[ ]*$/', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            unset($data['phone_number']);
        }
    }

    echo json_encode($this->model->search($data));
}

/**
 * Контроллер для відображення сторінки редагування клієнта
 *
 * @param $params - параметри
 * @param $data - дані
 * @return mixed
 */
public function action_edit($params, $data)
{
    $data['user_id'] = (int)$params;
    $view = "edit";
    $data['title'] = 'Клієнти · ' . $this->config-
>projectName;

    $data['user'] = $this->model-
>get_client($data['user_id']);
    $data['order'] = $this->model-
>get_order($data['user_id']);

    $data['user_name'] = $data['user']['first_name'] . ' ' .
    $data['user']['last_name'] . ' ' .
    $data['user']['fathers_name'];

```

```

        $this->view->out($view, $data);
    }

    /**
     * Контроллер для оновлення даних клієнта
     *
     * @param $params - параметри
     * @param $data - дані
     * @return mixed
     */
    public function action_update($params, $data)
    {
        $data['user_id'] = (int)urldecode($this->request->post->id);

        $data['first_name'] = urldecode($this->request->post->first_name);
        $data['last_name'] = urldecode($this->request->post->last_name);
        $data['father_name'] = urldecode($this->request->post->father_name);
        $data['phone_number'] = urldecode($this->request->post->phone_number);
        $data['discount'] = (int)urldecode($this->request->post->discount);

        if (strlen($data['first_name']) < 3 ||
            strlen($data['first_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
        }

        if (strlen($data['last_name']) < 3 ||
            strlen($data['last_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
        }
    }

```

```

        if (strlen($data['father_name']) < 3 ||
strlen($data['father_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
        }

        preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([
]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[
]*$/ ', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        $matches = array_slice($matches, 1);
        for ($i = 0; $i < count($matches); $i++) {
            $matches[$i] = (int)$matches[$i][0];
        }
        $data['phone_number'] = (int)implode($matches);

        if ($data['discount'] < 0 || $data['discount'] > 100) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($this->model->update($data)) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }
}

```

Файл app/admin/client/model.php

<?

```

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

```

```

class adminClientModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для отримання списку клієнтів
     *
     * @return array
     */
    public function create($data)
    {
        if ($data['discount'] == 0) {
            $data['discount'] = 'NULL';
        }

        $sql = 'INSERT INTO "client" (first_name, last_name,
fathers_name, phone_number, discount) VALUES (\'' . $this->db-
>escape($data['first_name']) . '\', \'' . $this->db-
>escape($data['last_name']) . '\', \'' . $this->db-
>escape($data['father_name']) . '\', ' . $this->db-
>escape((string)$data['phone_number']) . ', ' . $this->db-
>escape((string)$data['discount']) . ');';

        $this->db->query($sql);
        return !$this->db->last_error;
    }
}

```



```

}

/**
 * Метод моделі для пошуку працівників
 *
 * @param $data - дані
 * @return array
 */
public function search($data)
{
    foreach ($data as $key => $value) {
        $data[$key] = $this->db->escape($value);
    }

    $sql = 'SELECT * FROM client';
    if (isset($data['first_name']) ||
        isset($data['last_name']) || isset($data['father_name']) ||
        isset($data['phone_number'])) {
        $sql .= ' WHERE ';
    }
    if (isset($data['first_name'])) {
        $sql .= 'client.first_name LIKE \'%' .
            $data['first_name'] . '%\'';
    }
    if (isset($data['last_name'])) {
        $sql .= ' AND client.last_name LIKE \'%' .
            $data['last_name'] . '%\'';
    }
    if (isset($data['father_name'])) {
        $sql .= ' AND client.fathers_name LIKE \'%' .
            $data['father_name'] . '%\'';
    }
    if (isset($data['phone_number'])) {
        $sql .= ' AND client.phone_number = ' .
            $data['phone_number'];
    }
}

```

```

        $sql .= ' ';

        $result = (array)$this->db->query($sql);
        if ($result) {
            return $result;
        } else {
            return [];
        }
    }

    /**
     * Метод моделі для отримання даних акаунту клієнта
     *
     * @param $id - id клієнта
     * @return array
     */
    public function get_client($id)
    {
        $sql = 'SELECT * FROM client WHERE id = ' . $this->db->escape($id) . ' ';
        $result = (array)$this->db->query($sql);
        if ($result) {
            return $result[0];
        } else {
            return [];
        }
    }

    /**
     * Метод моделі для отримання списку замовлень клієнта
     *
     * @param $id - id клієнта
     * @return array
     */
    public function get_order($id)
    {

```

```

        $sql = 'SELECT "order".id, service.service_name,
"order".completed, "order".discount, "order".payment, car.make,
car.model, car.license_plate, service.price, "order".create_date
FROM task, "order", car, service WHERE task.id_order =
"order".id AND "order".id_car = car.id AND task.id_service =
service.id AND "order".id_client = 1;';

        $result = (array)$this->db->query($sql);
        if ($result) {
            $data = [];
            foreach ($result as $row) {
                if (isset($data[$row['id']])) {
                    $data[$row['id']]['services'] .= ', ' .
$row['service_name'];
                    $data[$row['id']]['price'] += $row['price'];
                } else {
                    $data[$row['id']] = [
                        'id' => $row['id'],
                        'date' => $row['create_date'],
                        'completed' => $row['completed'],
                        'discount' => $row['discount'],
                        'payment' => $row['payment'],
                        'price' => $row['price'],
                        'car' => implode(' - ', [$row['make'],
$row['model'], $row['license_plate']]),
                        'services' => $row['service_name']
                    ];
                }
            }
            return $data;
        } else {
            return [];
        }
    }

    /**
     * Метод моделі для оновлення даних клієнта

```

```

*
* @param $data - дані
* @return bool
*/
public function update($data)
{
    if ($data['discount'] == 0) {
        $data['discount'] = 'NULL';
    }

    if ($data['father_name'] == '') {
        $sql = 'UPDATE client SET first_name = \'' . $this->db->escape($data['first_name']) . '\', last_name = \'' . $this->db->escape($data['last_name']) . '\', fathers_name = NULL,
phone_number = ' . $this->db->escape((string)$data['phone_number']) . ', discount = ' .
$this->db->escape((string)$data['discount']) . ' WHERE id = ' .
$this->db->escape($data['user_id']) . ' ';
    } else {
        $sql = 'UPDATE client SET first_name = \'' . $this->db->escape($data['first_name']) . '\', last_name = \'' . $this->db->escape($data['last_name']) . '\', fathers_name = \'' .
$this->db->escape($data['father_name']) . '\', phone_number = ' .
$this->db->escape((string)$data['phone_number']) . ', discount
= ' . $this->db->escape((string)$data['discount']) . ' WHERE id
= ' . $this->db->escape($data['user_id']) . ' ';
    }

    $this->db->query($sql);
    return !$this->db->last_error;
}
}

```

Файл app/admin/client/view_edit.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
  <meta charset="utf-8">
  <meta name="author" content="Dmytro Zamkovyi">
  <title><?= $title ?></title>
  <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
  <header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">

```

```

        <a href="/admin/client/index" class="d-flex align-items-
center mb-3 mb-md-0 me-md-auto link-body-emphasis text-
decoration-none" style="margin-left: 16px;">
            
            <span class="fs-4">Назад</span>
        </a>

    <h4 class="d-flex align-items-center mb-3 mb-md-0 me-md-
auto link-body-emphasis text-decoration-none"><?= $user_name
?></h4>

    <button id="escape" type="button" class="btn btn-
outline-secondary" style="margin: 0 8px;">Скасувати</button>
    <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;">Зберегти</button>
</header>

<main>
    <div class="container">
        <div class="input-group mb-3">
            <span class="input-group-text">ПІБ</span>
            <input id="first-name" type="text" class="form-
control" placeholder="Прізвище" value="<?= $user['first_name']
?>">

            <input id="last-name" type="text" class="form-
control" placeholder="Ім'я" value="<?= $user['last_name'] ?>">
            <input id="father-name" type="text" class="form-
control" placeholder="Патронім" value="<?= $user['fathers_name']
?>">

        </div>

        <div class="input-group mb-3">
            <span class="input-group-text">Номер
телефону</span>

```

```

        <input id="phone-number" type="text"
class="form-control" value="<?= $user['phone_number'] ?>">
        <label class="input-group-text" for="phone-
number">+ZZZ (YY) XXX-XX-XX</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Знижка</span>
        <input id="discount" type="text" class="form-
control" value="<?= (int)$user['discount'] ?>">
        <span class="input-group-text">%</span>
    </div>
</div>

<div style="margin-left: 16px; margin-right: 16px;">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Автомобіль</th>
                <th scope="col">Дата</th>
                <th scope="col">Послуги</th>
                <th scope="col">Ціна, без знижки</th>
                <th scope="col">Знижка</th>
                <th scope="col">Сума</th>
                <th scope="col">Чи оплачено</th>
                <th scope="col">Чи виконано</th>
            </tr>
        </thead>
        <tbody class="table-group-divider">
            <? $i = 0; ?>
            <? foreach ($order as $row) : ?>
                <? $i++; ?>
                <tr>
                    <th scope="row"><?= $i ?></th>
                    <td><?= $row['car'] ?></td>

```

```

        <td><?= $row['date'] ?></td>
        <td><?= $row['services'] ?></td>
        <td><?= $row['price'] ?></td>
        <? if ($row['discount'] == 0) : ?>
            <td>-</td>
        <? else : ?>
            <td><?= $row['discount']
?>%</td>

        <? endif; ?>
        <td><?= round($row['price'] * (1 -
$row['discount'] / 100), 2) ?></td>
        <? if ($row['payment'] == 'f') : ?>
            <td><input class="form-check-
input" type="checkbox" disabled></td>
        <? else : ?>
            <td><input class="form-check-
input" type="checkbox" checked disabled></td>
        <? endif; ?>
        <? if ($row['completed'] == 'f') :
?>
            <td><input class="form-check-
input" type="checkbox" disabled></td>
        <? else : ?>
            <td><input class="form-check-
input" type="checkbox" checked disabled></td>
        <? endif; ?>
        <td><button type="button"
class="edit btn" style="padding: 0;" onclick=edit(<?= $row['id']
?>)></button></td>
    </tr>
    <? endforeach; ?>
</tbody>
</table>
</div>
</main>

```



```

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const btn_cancel =
document.getElementById('escape');
        const btn_save = document.getElementById('save');
        const inp_first_name =
document.getElementById('first-name');
        const inp_last_name = document.getElementById('last-
name');
        const inp_father_name =
document.getElementById('father-name');
        const inp_phone_number =
document.getElementById('phone-number');
        const inp_discount =
document.getElementById('discount');

        /**
         * Функція для перенаправлення на сторінку зі
СПИСКОМ клієнтів
         *
         * @returns {void}
         */
        btn_cancel.addEventListener('click', () => {
            window.location.href = '/admin/client/index';
        });

        /**
         * Функція для збереження даних клієнта
         *
         * @returns {void}
         */
        btn_save.addEventListener('click', () => {
            let is_error = false;
            const data = {
                id: <?= $user['id'] ?>,

```

```

        first_name: inp_first_name.value,
        last_name: inp_last_name.value,
        father_name: inp_father_name.value,
        phone_number: inp_phone_number.value,
        discount: parseInt(inp_discount.value)
    };

    if (data.first_name.length < 3 ||
data.first_name.length > 40) {
        is_error = true;
        inp_first_name.classList.add('is-invalid');
    } else {
        inp_first_name.classList.remove('is-
invalid');
    }

    if (data.last_name.length < 3 ||
data.last_name.length > 40) {
        is_error = true;
        inp_last_name.classList.add('is-invalid');
    } else {
        inp_last_name.classList.remove('is-
invalid');
    }

    if (data.father_name != '' &&
data.father_name.length < 3 || data.father_name.length > 40) {
        is_error = true;
        inp_father_name.classList.add('is-invalid');
    } else {
        inp_father_name.classList.remove('is-
invalid');
    }

    if (data.phone_number == '') {

```

```

        inp_phone_number.classList.add('is-
invalid');

        is_error = true;
    } else {
        inp_phone_number.classList.remove('is-
invalid');

        let re = /^[ ]*\+?([0-9]{2})[-\(\ ]*([0-
9]{1})[-\(\ ]*([0-9]{2})[-\)\ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-
9]{2})[ ]*$/gm;

        let res = re.exec(data.phone_number);
        if (res) {
            data.phone_number =
res.slice(1).join('');
        } else {
            inp_phone_number.classList.add('is-
invalid');

            is_error = true;
        }
    }

    if (data.discount == '') {
        data.discount = null;
    } else {
        inp_discount.classList.remove('is-invalid');
        if (data.discount < 0 || data.discount >
100) {

            inp_discount.classList.add('is-
invalid');

            is_error = true;
        }
    }

    if (is_error) {
        return;
    }

```

```

        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/admin/client/update', true);
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                btn_save.classList.remove('btn-danger');
                btn_save.classList.add('btn-success');
                btn_save.innerHTML = 'Збережено';
            } else {
                btn_save.classList.remove('btn-
success');

                btn_save.classList.add('btn-danger');
                btn_save.innerHTML = 'Помилка';
            }
        };
        xhr.send(post_str(data));
    });
});

/**
 * Функція для перетворення об'єкту в строку
 *
 * @param {object} data - об'єкт з даними
 * @returns {string} - строка з даними
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

```

```

        /**
        * Функція для перенаправлення на сторінку редагування
замовлення
        *
        * @param {int} id - id замовлення
        * @returns {void}
        */
        function edit(id) {
            window.location.href = '/admin/order/edit/' + id;
        }
    </script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/admin/client/view_index.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.

```

- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

<head>

<meta charset="utf-8">

<meta name="author" content="Dmytro Zamkovyi">

<title><?= \$title ?></title>

<link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>

<?= \$head ?>

<main>

<div class="container">

<div style="margin-top: 16px;">

<button id="new-client" type="button" class="btn
btn-primary" data-bs-toggle="modal" data-bs-target="#add-
client">Новий клієнт</button>

<output hidden id="success" class="text-
success">Користувача успішно додано!</output>

<div class="modal fade" id="add-client" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-
centered modal-lg modal-dialog-scrollable">

<div class="modal-content">

```

        <div class="modal-header">
            <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту клієнта</h1>
            <button type="button"
class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
        </div>
        <div class="modal-body">
            <div class="input-group mb-3">
                <span class="input-group-
text">ПІБ</span>
                <input id="first-name"
type="text" class="form-control" placeholder="Прізвище">
                <input id="last-name"
type="text" class="form-control" placeholder="Ім'я">
                <input id="father-name"
type="text" class="form-control" placeholder="Патронім">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-
text">Номер телефону</span>
                <input id="phone-number"
type="text" class="form-control">
                <label class="input-group-
text" for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
            </div>
            <div class="input-group mb-3">
                <span class="input-group-
text">Знижка (опціонально)</span>
                <input id="discount"
type="text" class="form-control">
                <span class="input-group-
text"></span>

```

```

                                <span class="input-group-
text">0.00</span>

                                </div>
                            </div>
                            <div class="modal-footer">
                                <output hidden id="error"
class="text-danger"></output>
                                <button type="button" class="btn
btn-secondary" data-bs-dismiss="modal">Відміна</button>
                                <button id="create"
type="button" class="btn btn-primary">Створити</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

            <hr>

            <div class="accordion" id="according-search">
                <div class="accordion-item">
                    <h2 class="accordion-header">
                        <button class="accordion-button
collapsed" type="button" data-bs-toggle="collapse" data-bs-
target="#collapse-search" aria-expanded="false" aria-
controls="collapseTwo">
                            <h1>Пошук</h1>
                        </button>
                    </h2>
                    <div id="collapse-search" class="accordion-
collapse collapse" data-bs-parent="#according-search">
                        <div class="accordion-body">
                            <div class="input-group mb-3">
                                <span class="input-group-
text">ПІБ</span>

```



```

        <input id="search-first-name"
type="text" class="form-control" placeholder="Прізвище">
        <input id="search-last-name"
type="text" class="form-control" placeholder="Ім'я">
        <input id="search-father-name"
type="text" class="form-control" placeholder="Патронім">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Номер телефону</span>
        <input id="search-phone-number"
type="text" class="form-control">
        <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
    </div>

    <div class="gap-2 d-md-flex justify-
content-md-end">
        <button id="search"
type="button" class="btn btn-primary">Пошук</button>
        <button id="search-reset"
type="button" class="btn btn-outline-primary">Скинути</button>
    </div>
</div>
</div>
</div>
<div style="margin-top: 16px;">
    <button id="reload" type="button" class="btn
btn-outline-success ms-auto">Оновити</button>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">ПІБ</th>

```

```

        <th scope="col">Телефон</th>
        <th scope="col">Знижка</th>
    </tr>
</thead>
<tbody id="table-body" class="table-group-
divider"></tbody>
</table>
</div>
</main>

<script>
    let client = Object();
    window.addEventListener("DOMContentLoaded", () => {
        const modal = new
bootstrap.Modal(document.getElementById('add-client'));
        const btn_new_client = document.getElementById("new-
client");
        const btn_create =
document.getElementById("create");
        const success = document.getElementById("success");
        const error = document.getElementById("error");
        const inp_first_name =
document.getElementById("first-name");
        const inp_last_name = document.getElementById("last-
name");
        const inp_father_name =
document.getElementById("father-name");
        const inp_phone_number =
document.getElementById("phone-number");
        const inp_discount =
document.getElementById("discount");

        const inp_search_first_name =
document.getElementById("search-first-name");
        const inp_search_last_name =
document.getElementById("search-last-name");

```

```

        const inp_search_father_name =
document.getElementById("search-father-name");
        const inp_search_phone_number =
document.getElementById("search-phone-number");
        const btn_search =
document.getElementById("search");
        const btn_search_reset =
document.getElementById("search-reset");
        const btn_reload =
document.getElementById("reload");
        const table_body = document.getElementById("table-
body");

        send();

/**
 * Функція для очищення полів форми при відкритті
модального вікна
 *
 * @returns {void}
 */
btn_new_client.addEventListener("click", () => {
    success.hidden = true;
    error.hidden = true;
    inp_first_name.value = "";
    inp_last_name.value = "";
    inp_father_name.value = "";
    inp_phone_number.value = "";
    inp_discount.value = "";
});

/**
 * Функція для створення нового користувача
 *
 * @returns {void}
 */

```

```

btn_create.addEventListener("click", () => {
    let is_error = false;
    const data = {
        first_name: inp_first_name.value,
        last_name: inp_last_name.value,
        father_name: inp_father_name.value,
        phone_number: inp_phone_number.value,
        discount: parseInt(inp_discount.value)
    };

    if (data.first_name.length < 3 ||
data.first_name.length > 40) {
        inp_first_name.classList.add("is-invalid");
        is_error = true;
    } else {
        inp_first_name.classList.remove("is-
invalid");
    }

    if (data.last_name.length < 3 ||
data.last_name.length > 40) {
        inp_last_name.classList.add("is-invalid");
        is_error = true;
    } else {
        inp_last_name.classList.remove("is-
invalid");
    }

    if (data.father_name.length < 3 ||
data.father_name.length > 40) {
        inp_father_name.classList.add("is-invalid");
        is_error = true;
    } else {
        inp_father_name.classList.remove("is-
invalid");
    }
}

```

```

        if (data.phone_number == '') {
            inp_phone_number.classList.add('is-
invalid');

            inp_phone_number.placeholder = 'Введіть
номер телефону'

            is_error = true;
        } else {
            inp_phone_number.classList.remove('is-
invalid');

            let re = /^[ ]*\+?([0-9]{2})[-\([ ]*([0-
9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-
9]{2})[ ]*$/gm;

            let res = re.exec(data.phone_number);
            if (res) {
                data.phone_number =
res.slice(1).join('');
            } else {
                inp_phone_number.classList.add('is-
invalid');

                inp_phone_number.placeholder = 'Невірний
формат номеру телефону'

                is_error = true;
            }
        }

        if (data.discount < 0 || data.discount > 100) {
            inp_discount.classList.add("is-invalid");
            is_error = true;
        } else {
            if (data.discount == '') {
                delete data.discount;
            }

            inp_discount.classList.remove("is-invalid");
        }
    }

```

```

        if (is_error) {
            error.hidden = false;
            error.innerHTML = "Перевірте правильність
введених даних!";
            return;
        }

        let xhr = new XMLHttpRequest();
        xhr.open("POST", "/admin/client/create");
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                success.hidden = false;
                error.hidden = true;
                modal.hide();

                setTimeout(() => {
                    success.hidden = true;
                }, 10000);

                search();
            } else {
                error.hidden = false;
                error.innerHTML = "ERROR: Неможливо
створити користувача";
            }
        };
        xhr.send(post_str(data));

    });

/**
 * Функція скидування параметрів пошуку

```

```

*
* @param {Number} id - id користувача
* @returns {void}
*/
btn_search_reset.addEventListener("click", () => {
    inp_search_first_name.value = "";
    inp_search_last_name.value = "";
    inp_search_father_name.value = "";
    inp_search_phone_number.value = "";
    search();
});

btn_reload.addEventListener("click", send);
btn_search.addEventListener("click", search);

/**
 * Функція для пошуку користувачів
 *
 * @returns {void}
 */
function search() {
    let data = {
        first_name: inp_search_first_name.value,
        last_name: inp_search_last_name.value,
        father_name: inp_search_father_name.value,
        phone_number: inp_search_phone_number.value
    };

    if (data.first_name == '') {
        delete data.first_name;
    }

    if (data.last_name == '') {
        delete data.last_name;
    }
}

```

```

        if (data.father_name == '') {
            delete data.father_name;
        }

        if (data.phone_number == '') {
            delete data.phone_number;
        }

        if (data.phone_number == '') {
            delete data.phone_number;
        }

        send(data);
    }

/**
 * Функція для відправки запиту
 *
 * @param {Array} inp_data - дані для відправки
 * @returns void
 */
function send(inp_data = Array()) {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/client/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status != 200) {
            document.getElementById('table_body').innerHTML = '';
        } else {
            document.getElementById('error').hidden
= true;

```



```

        let result =
JSON.parse(xhr.responseText);
        client = result;
        table_body.innerHTML = '';
        for (let i = 0; i < result.length; i++)
{
            let tr =
document.createElement('tr');
            tr.classList.add('col-12', 'col-lg-
auto', 'my-2', 'justify-content-center', 'my-md-0');
            tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>'
            tr.innerHTML += '<td>' + sum_strs('
', result[i]['first_name'], result[i]['last_name'],
result[i]['fathers_name']) + '</td>'
            tr.innerHTML += '<td>' +
result[i]['phone_number'] + '</td>'
            if (result[i]['discount'] == null) {
                result[i]['discount'] = '-';
            }
            tr.innerHTML += '<td>' +
result[i]['discount'] + '</td>'
            tr.innerHTML += '<td>' + '<button
type="button" class="edit btn" style="padding: 0;"
onclick=edit(' + result[i]['id'] + ')></button>'
+ '</td>'
            table_body.appendChild(tr);
        }
    }
};
xhr.send(post_str(inp_data));
}
});

/**

```

```

* Функція для перетворення об'єкту в строку
*
* @param {Object} data - об'єкт
* @returns {String} - строка
*/
function post_str(data) {
    let str = '';
    for (let key in data) {
        if (data[key] == undefined) {
            continue;
        }
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

/**
* Функція для склеювання строк
*
* @param {String} split - розділювач
* @param {...String} theArgs - строки
* @returns {String} - склеєна строка
*/
function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

/**

```

```

        * Функція для перенаправлення на сторінку редагування
        *
        * @param {Number} id - id клієнта
        * @returns {void}
        */
function edit(id) {
    window.location.href = '/admin/client/edit/' + id;
}
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл `app/admin/employee/action.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.

```

```

*/

class adminEmployee extends Curswork\ActionClass
{
    /**
     * Контроллер для відображення списку працівників
     *
     * @param $params - параметри
     * @param $data - дані
     * @return mixed
     */
    public function action_index($params, $data)
    {
        $view = "index";
        $data['label'] = 'Перегляд працівників';
        $data['menu'] = array(
            array(
                'name' => 'Працівники',
                'href' => '/admin/employee/index',
                'active' => true,
                'img' => '/data/icon/employee_activ.png'
            ),
            array(
                'name' => 'Послуги',
                'href' => '/admin/service/index',
                'active' => false,
                'img' => '/data/icon/service.png'
            ),
            array(
                'name' => 'Замовлення',
                'href' => '/admin/order/index',
                'active' => false,
                'img' => '/data/icon/order.png'
            ),
            array(
                'name' => 'Клієнти',

```

```

        'href' => '/admin/client/index',
        'active' => false,
        'img' => '/data/icon/client.png'
    ),
    array(
        'name' => 'Вихід',
        'href' => '/index/index/logout',
        'active' => false,
        'img' => '/data/icon/logout.png'
    ),
);
$data['head'] = $this->load('index', 'index', 'head',
'', $data);
$data['title'] = 'Працівники · ' . $this->config-
>projectName;

$data['role_list'] = $this->model->get_role_list();
$data['sto_list'] = $this->model->get_sto_list();
$data['branch_list'] = $this->model->get_branch_list();
$data['position_list'] = $this->model-
>get_position_list();

$this->view->out($view, $data);
}

/**
 * Контроллер для додаання нового працівника
 *
 * @param $params - параметри
 * @param $data - дані
 */
public function action_create($params, $data)
{
    $data['role'] = urldecode($this->request->post->role);
    $data['sto'] = urldecode($this->request->post->sto);

```

```

        $data['branch'] = urldecode($this->request->post->branch);
        $data['position'] = urldecode($this->request->post->position);
        $data['first_name'] = urldecode($this->request->post->first_name);
        $data['last_name'] = urldecode($this->request->post->last_name);
        $data['father_name'] = urldecode($this->request->post->father_name);
        $data['phone_number'] = urldecode($this->request->post->phone_number);
        $data['login'] = urldecode($this->request->post->login);
        $data['pwd'] = urldecode($this->request->post->pwd);
        $data['tarif'] = urldecode($this->request->post->tarif);

        preg_match_all('/^[ ]*\+?([0-9]{2})[-\(\ ]*([0-9]{1})[-\(\ ]*([0-9]{2})[-\(\ ]*([0-9]{3})[-\(\ ]*([0-9]{2})[-\(\ ]*([0-9]{2})[ ]*$/ ', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        $matches = array_slice($matches, 1);
        for ($i = 0; $i < count($matches); $i++) {
            $matches[$i] = (int)$matches[$i][0];
        }
        $data['phone_number'] = (int)implode($matches);

        $data['role'] = (int)$data['role'];
        $data['sto'] = $this->model->get_sto_id($data['sto']);
        $data['branch'] = $this->model->get_branch_id($data['branch']);
        $data['position'] = $this->model->get_position_id($data['position']);
        $data['tarif'] = (float)$data['tarif'];

```

```

        $data['pwd'] = hash('md5', $data['pwd']);

        if ($this->model->create($data)) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }

    /**
     * Контроллер для пошуку працівників
     *
     * @param $params - параметри
     * @param $data - дані
     * @return mixed
     */
    public function action_search($params, $data)
    {
        if (isset($this->request->post->role)) {
            $data['role'] = (int)urldecode($this->request->post->role);
        }

        if (isset($this->request->post->sto)) {
            $data['sto'] = (int)$this->model->get_sto_id(urldecode($this->request->post->sto));
        }

        if (isset($this->request->post->branch)) {
            $data['branch'] = (int)$this->model->get_branch_id(urldecode($this->request->post->branch));
        }

        if (isset($this->request->post->position)) {
            $data['position'] = (int)$this->model->get_position_id(urldecode($this->request->post->position));
        }

        if (isset($this->request->post->first_name)) {

```

```

        $data['first_name'] = urldecode($this->request->post->first_name);
    }
    if (isset($this->request->post->last_name)) {
        $data['last_name'] = urldecode($this->request->post->last_name);
    }
    if (isset($this->request->post->father_name)) {
        $data['father_name'] = urldecode($this->request->post->father_name);
    }
    if (isset($this->request->post->login)) {
        $data['login'] = urldecode($this->request->post->login);
    }

    $result = $this->model->search($data);

    if (!$this->db->last_error) {
        header('HTTP/1.1 200 OK');
        echo json_encode($result);
    } else {
        header('HTTP/1.1 400 Bad Request');
    }
}

/**
 * Контроллер для відображення сторінки редагування
працівника
 *
 * @param $params - параметри
 * @param $data - дані
 * @return mixed
 */
public function action_edit($params, $data)
{

```



```

        $data['user_id'] = (int)$params;
        $view = "edit";
        $data['title'] = 'Працівники · ' . $this->config-
>projectName;

        $data['user_name'] = $this->model-
>get_account($data['user_id']);

        $data['role_list'] = $this->model->get_role_list();
        $data['sto_list'] = $this->model->get_sto_list();
        $data['branch_list'] = $this->model->get_branch_list();
        $data['position_list'] = $this->model-
>get_position_list();
        $data['employee'] = $this->model-
>get_employee($data['user_id']);

        $this->view->out($view, $data);
    }

    /**
     * Контроллер для оновлення працівника
     *
     * @param $params - параметри
     * @param $data - дані
     */
    public function action_update($params, $data)
    {
        $data['user_id'] = (int)$params;
        $data['role'] = urldecode($this->request->post->role);
        $data['sto'] = urldecode($this->request->post->sto);
        $data['branch'] = urldecode($this->request->post-
>branch);
        $data['position'] = urldecode($this->request->post-
>position);
        $data['first_name'] = urldecode($this->request->post-
>first_name);

```

```

        $data['last_name'] = urldecode($this->request->post->last_name);
        $data['father_name'] = urldecode($this->request->post->father_name);
        $data['phone_number'] = urldecode($this->request->post->phone_number);
        $data['login'] = urldecode($this->request->post->login);
        $data['pwd'] = urldecode($this->request->post->pwd);
        $data['tarif'] = urldecode($this->request->post->tarif);

        preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        $matches = array_slice($matches, 1);
        for ($i = 0; $i < count($matches); $i++) {
            $matches[$i] = (int)$matches[$i][0];
        }
        $data['phone_number'] = (int)implode($matches);

        $data['role'] = (int)$data['role'];
        $data['sto'] = $this->model->get_sto_id($data['sto']);
        $data['branch'] = $this->model->get_branch_id($data['branch']);
        $data['position'] = $this->model->get_position_id($data['position']);
        $data['tarif'] = (float)$data['tarif'];
        if ($data['pwd'] != '') {
            $data['pwd'] = hash('md5', $data['pwd']);
        } else {
            unset($data['pwd']);
        }
    }

```

```

        if ($this->model->update($data)) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }

}

/**
 * Контроллер для видалення працівника
 *
 * @param $params - параметри
 * @param $data - дані
 * @return mixed
 */
public function action_delete($params, $data)
{
    $data['user_id'] = (int)$params;
    $this->model->delete($data['user_id']);
    header('HTTP/1.1 200 OK');
    header('Location: /admin/employee/index');
}
}

```

Файл `app/admin/employee/model.php`

<?

```

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.

```

```

* - Copying or reproducing any part of this code in any form.
* - Modifying or altering any part of this code.
*
* See the full text of the license in the LICENSE file in the
root directory of this repository.
*/

```

```

class adminEmployeeModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для отримання списку ролей
     *
     * @return array
     */
    public function get_role_list(): array
    {
        $result = (array)$this->db->query('SELECT * FROM role');
        if ($result) {
            for ($i = 0; $i < count($result); $i++) {
                if ($result[$i]['role_name'] == 'employee') {
                    $result[$i]['active'] = true;
                } else {
                    $result[$i]['active'] = false;
                }
            }
            return $result;
        } else {
            return array();
        }
    }

    /**
     * Метод моделі для отримання списку СТО
     *
     * @return array
     */
}

```

```

public function get_sto_list(): array
{
    $result = (array)$this->db->query('SELECT id, sto_name
FROM sto');
    if ($result) {
        return $result;
    } else {
        return array();
    }
}

/**
 * Метод моделі для отримання списку відділень
 *
 * @return array
 */
public function get_branch_list(): array
{
    $result = (array)$this->db->query('SELECT id,
department_name FROM branch');
    if ($result) {
        return $result;
    } else {
        return array();
    }
}

/**
 * Метод моделі для отримання списку посад
 *
 * @return array
 */
public function get_position_list(): array
{
    $result = (array)$this->db->query('SELECT id,
position_name FROM position');

```

```

        if ($result) {
            return $result;
        } else {
            return array();
        }
    }
}

/**
 * Метод моделі для id сто по назві
 *
 * @param $sto_name - назва сто
 * @return bool|array
 */
public function get_sto_id($sto_name): bool|array
{
    $result = (array)$this->db->query('SELECT id FROM sto
WHERE sto_name = \'' . $this->db->escape($sto_name) . '\';');
    if ($result) {
        return (int)$result[0]['id'];
    } else {
        return false;
    }
}

/**
 * Метод моделі для id відділення по назві
 *
 * @param $branch_name - назва відділення
 * @return bool|array
 */
public function get_branch_id($branch_name): bool|array
{
    $result = (array)$this->db->query('SELECT id FROM branch
WHERE department_name = \'' . $this->db->escape($branch_name) .
'\';');
    if ($result) {

```

```

        return (int)$result[0]['id'];
    } else {
        return false;
    }
}

/**
 * Метод моделі для id посади по назві
 *
 * @param $position_name - назва посади
 * @return bool|array
 */
public function get_position_id($position_name): bool|array
{
    $result = (array)$this->db->query('SELECT id FROM
position WHERE position_name = \'' . $this->db-
>escape($position_name) . '\\';');
    if ($result) {
        return (int)$result[0]['id'];
    } else {
        return false;
    }
}

/**
 * Метод моделі для створення працівника
 *
 * @param $data - дані
 * @return bool
 */
public function create($data): bool
{
    foreach ($data as $key => $value) {
        $data[$key] = $this->db->escape($value);
    }
}

```

```

        $sql = 'INSERT INTO employee (id_role, id_sto,
id_branch, id_position, first_name, last_name, fathers_name,
phone_number, login, password, tariff) VALUES (' . $data['role']
. ', ' . $data['sto'] . ', ' . $data['branch'] . ', ' .
$data['position'] . ', \'' . $data['first_name'] . '\', \'' .
$data['last_name'] . '\', \'' . $data['father_name'] . '\', ' .
$data['phone_number'] . ', \'' . $data['login'] . '\', \'' .
$data['pwd'] . '\', ' . $data['tarif'] . ');';

        $this->db->query($sql);

        if ($this->db->last_error) {
            return false;
        } else {
            return true;
        }
    }

}

/**
 * Метод моделі для пошуку працівників
 *
 * @param $data - дані
 * @return array
 */
public function search($data): array
{
    foreach ($data as $key => $value) {
        $data[$key] = $this->db->escape($value);
    }

    $sql = 'SELECT employee.id AS employee_id,
employee.phone_number AS employee_pn, * FROM employee, role,
sto, branch, position';

    $sql .= ' WHERE';

    $sql .= ' employee.id_role = role.id AND';

    $sql .= ' employee.id_sto = sto.id AND';

    $sql .= ' employee.id_branch = branch.id AND';

    $sql .= ' employee.id_position = position.id';

```



```

        if (isset($data['first_name'])) {
            $sql .= ' AND employee.first_name LIKE \'%' .
$data['first_name'] . '%\'';
        }
        if (isset($data['last_name'])) {
            $sql .= ' AND employee.last_name LIKE \'%' .
$data['last_name'] . '%\'';
        }
        if (isset($data['father_name'])) {
            $sql .= ' AND employee.fathers_name LIKE \'%' .
$data['father_name'] . '%\'';
        }
        if (isset($data['login'])) {
            $sql .= ' AND employee.login = \'' . $data['login']
. '\'';
        }
        if (isset($data['role'])) {
            $sql .= ' AND employee.id_role = ' . $data['role'];
        }
        if (isset($data['sto'])) {
            $sql .= ' AND employee.id_sto = ' . $data['sto'];
        }
        if (isset($data['branch'])) {
            $sql .= ' AND employee.id_branch = ' .
$data['branch'];
        }
        if (isset($data['position'])) {
            $sql .= ' AND employee.id_position = ' .
$data['position'];
        }

        $sql .= ';';

        $result = (array)$this->db->query($sql);

```

```

        for ($i = 0; $i < count($result); $i++) {
            unset($result[$i]['password']);
        }
        return $result;
    }

/**
 * Метод моделі для отримання працівника
 *
 * @param $id - id працівника
 * @return array
 */
public function get_employee($id): array
{
    $result = (array)$this->db->query('SELECT employee.id AS
employee_id, employee.phone_number AS employee_pn, * FROM
employee, role, sto, branch, position WHERE employee.id = ' .
$id . ' AND employee.id_role = role.id AND employee.id_sto =
sto.id AND employee.id_branch = branch.id AND
employee.id_position = position.id;');
    if ($result) {
        unset($result[0]['password']);
        return $result[0];
    } else {
        return false;
    }
}

/**
 * Метод моделі для отримання імені працівника
 *
 * @param $user_id - id працівника
 * @return string
 */
public function get_account($user_id): string
{

```

```

        $res = $this->db->query('SELECT first_name, last_name,
fathers_name FROM employee WHERE id=' . $user_id . ';' );
        if ($res) {
            return trim(implode(' ', $res[0]));
        }
        return '';
    }

    /**
     * Метод моделі для оновлення працівника
     *
     * @param $data - дані
     * @return bool
     */
    public function update($data): bool
    {
        foreach ($data as $key => $value) {
            $data[$key] = $this->db->escape($value);
        }

        if ($data['pwd'] != '') {
            $sql = 'UPDATE employee SET id_role = ' .
            $data['role'] . ', id_sto = ' . $data['sto'] . ', id_branch = '
            . $data['branch'] . ', id_position = ' . $data['position'] . ',
            first_name = \'' . $data['first_name'] . '\', last_name = \'' .
            $data['last_name'] . '\', fathers_name = \'' .
            $data['father_name'] . '\', phone_number = ' .
            $data['phone_number'] . ', login = \'' . $data['login'] . '\',
            password = \'' . $data['pwd'] . '\', tariff = ' . $data['tariff']
            . ' WHERE id = ' . $data['user_id'] . ';';
        } else {
            $sql = 'UPDATE employee SET id_role = ' .
            $data['role'] . ', id_sto = ' . $data['sto'] . ', id_branch = '
            . $data['branch'] . ', id_position = ' . $data['position'] . ',
            first_name = \'' . $data['first_name'] . '\', last_name = \'' .
            $data['last_name'] . '\', fathers_name = \'' .
            $data['father_name'] . '\', phone_number = ' .

```

```

$data['phone_number'] . ', login = \'' . $data['login'] . '\',
tariff = ' . $data['tariff'] . ' WHERE id = ' . $data['user_id']
. ';;';

    }

    $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    } else {
        return true;
    }
}

/**
 * Метод моделі для видалення працівника
 *
 * @param $id - id працівника
 * @return bool
 */
public function delete($id): bool
{
    $this->db->query('DELETE FROM employee WHERE id = ' .
$id . ';;');
    if ($this->db->last_error) {
        return false;
    } else {
        return true;
    }
}
}

```

Файл app/admin/employee/view_edit.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

```

```
<!--
```

```
License
```

```
Copyright (c) 2023 Dmytro Zamkovyi
```

```
You are prohibited from:
```

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

```
See the full text of the license in the LICENSE file in the root  
directory of this repository.
```

```
-->
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <meta name="author" content="Dmytro Zamkovyi">
```

```
    <title><?= $title ?></title>
```

```
    <link rel="shortcut icon" href="/data/favicon.ico"
```

```
type="image/png">
```

```
    <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot  
strap.min.css" rel="stylesheet" integrity="sha384-
```

```
T3c6CoIi6uLrA9TneNEoa7RxnatzjcdSCmGlMXxSRlGAsXEV/Dwwykc2MPK8M2HN  
" crossorigin="anonymous">
```

```
</head>
```

```
<body>
```

```
    <header class="d-flex flex-wrap justify-content-center py-3  
mb-4 border-bottom">
```

```
        <a href="/admin/employee/index" class="d-flex align-  
items-center mb-3 mb-md-0 me-md-auto link-body-emphasis text-  
decoration-none" style="margin-left: 16px;">
```

```

        
        <span class="fs-4">Назад</span>
    </a>

    <h4 class="d-flex align-items-center mb-3 mb-md-0 me-md-
auto link-body-emphasis text-decoration-none"><?= $user_name
?></h4>

    <button id="delete" type="button" class="btn btn-
outline-danger" style="margin: 0 8px;">Видалити</button>
    <button id="escape" type="button" class="btn btn-
outline-secondary" style="margin: 0 8px;">Скасувати</button>
    <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;">Зберегти</button>
</header>
<main class="container">
    <div class="modal-body">
        <div class="input-group mb-3">
            <label class="input-group-text"
for="role">Роль</label>
            <select class="form-select" id="role">
                <? foreach ($role_list as $role) : ?>
                    <? if ($employee['id'] == $role['id']) :
?>
                        <option value="<?= $role['id'] ?>"
selected><?= $role['role_name'] ?></option>
                    <? else : ?>
                        <option value="<?= $role['id']
?>"><?= $role['role_name'] ?></option>
                    <? endif ?>
                <? endforeach ?>
            </select>
        </div>

        <div class="input-group mb-3">

```

```

        <span class="input-group-text">ПІБ</span>
        <input id="first-name" type="text" class="form-
control" placeholder="Прізвище" value="<?=
$employee['first_name'] ?>">
        <input id="last-name" type="text" class="form-
control" placeholder="Ім'я" value="<?= $employee['last_name']
?>">
        <input id="father-name" type="text" class="form-
control" placeholder="Патронім" value="<?=
$employee['fathers_name'] ?>">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Номер
телефону</span>
        <input id="phone-number" type="text"
class="form-control" value="<?= $employee['employee_pn'] ?>">
        <label class="input-group-text" for="phone-
number">+ZZZ (YY) XXX-XX-XX</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Тариф</span>
        <input id="tarif" type="text" class="form-
control" value="<?= $employee['tariff'] ?>">
        <span class="input-group-text">€</span>
        <span class="input-group-text">0.00</span>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">СТО</span>
        <input id="sto" type="text" class="form-control"
list="sto-list" value="<?= $employee['sto_name'] ?>">
        <datalist id="sto-list">
            <? foreach ($sto_list as $sto) : ?>

```

```

        <option value="<?= $sto['sto_name']
?>"></option>

        <? endforeach ?>
    </datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Відділ</span>
    <input id="branch" type="text" class="form-
control" list="branch-list" value="<?=
$employee['department_name'] ?>">
    <datalist id="branch-list">
        <? foreach ($branch_list as $branch) : ?>
            <option value="<?=
$branch['department_name'] ?>"></option>
        <? endforeach ?>
    </datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Посада</span>
    <input id="position" type="text" class="form-
control" list="position-list" value="<?=
$employee['position_name'] ?>">
    <datalist id="position-list">
        <? foreach ($position_list as $position) :
?>
            <option value="<?=
$position['position_name'] ?>"></option>
        <? endforeach ?>
    </datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Лорін</span>

```



```

        <input id="login" type="email" class="form-
control" value="<?= $employee['login'] ?>">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Пароль</span>
        <input id="pwd" type="password" class="form-
control">
    </div>

    <p id="error" class="text-danger" hidden></p>
    <p id="success" class="text-success" hidden>Данні
успішно збережені</p>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const inp_role = document.getElementById('role');
        const inp_first_name =
document.getElementById('first-name');
        const inp_last_name = document.getElementById('last-
name');
        const inp_father_name =
document.getElementById('father-name');
        const inp_phone_number =
document.getElementById('phone-number');
        const inp_tarif = document.getElementById('tarif');
        const inp_sto = document.getElementById('sto');
        const inp_branch =
document.getElementById('branch');
        const inp_position =
document.getElementById('position');
        const inp_login = document.getElementById('login');
        const inp_pwd = document.getElementById('pwd');

```

```

        const btn_delete =
document.getElementById('delete');
        const btn_cancel =
document.getElementById('escape');
        const btn_save = document.getElementById('save');

/**
 * Функція для повернення на попередню сторінку
 *
 * @returns void
 */
btn_cancel.addEventListener('click', () => {
    window.location.href = '/admin/employee/index';
});

/**
 * Функція для видалення працівника
 *
 * @returns void
 */
btn_delete.addEventListener('click', () => {
    if (confirm('Ви дійсно хочете видалити цього
працівника?')) {
        window.location.href =
'/admin/employee/delete/' + '<?=$employee['employee_id'] ?>';
    }
});

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_save.addEventListener('click', () => {
    let is_error = false;
    let data = {

```

```

        role: inp_role.value,
        first_name: inp_first_name.value,
        last_name: inp_last_name.value,
        father_name: inp_father_name.value,
        phone_number: inp_phone_number.value,
        tarif: inp_tarif.value,
        sto: inp_sto.value,
        branch: inp_branch.value,
        position: inp_position.value,
        login: inp_login.value,
        pwd: inp_pwd.value
    };

    if (data.role == '') {
        inp_role.classList.add('is-invalid');
        inp_role.placeholder = 'Виберіть роль'
        is_error = true;
    } else {
        inp_role.classList.remove('is-invalid');
        data.role = data.role;
    }

    if (data.first_name.length < 3 ||
data.first_name.length > 40) {
        inp_first_name.classList.add('is-invalid');
        inp_first_name.placeholder = 'Введіть ім\'я'
        is_error = true;
    } else {
        inp_first_name.classList.remove('is-
invalid');
    }

    if (data.last_name.length < 3 ||
data.last_name.length > 40) {
        inp_last_name.classList.add('is-invalid');

```

```

        inp_last_name.placeholder = 'Введіть
прізвище'

        is_error = true;
    } else {
        inp_last_name.classList.remove('is-
invalid');
    }

    if (data.phone_number == '') {
        inp_phone_number.classList.add('is-
invalid');

        inp_phone_number.placeholder = 'Введіть
номер телефону'

        is_error = true;
    } else {
        inp_phone_number.classList.remove('is-
invalid');

        let re = /^[ ]*\+?([0-9]{2})[-\(\ ]*([0-
9]{1})[-\(\ ]*([0-9]{2})[-\(\ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-
9]{2})[ ]*$/gm;

        let res = re.exec(data.phone_number);
        if (res) {
            data.phone_number =
res.slice(1).join('');
        } else {
            inp_phone_number.classList.add('is-
invalid');

            inp_phone_number.placeholder = 'Невірний
формат номеру телефону'

            is_error = true;
        }
    }

    if (data.tarif == '') {
        inp_tarif.classList.add('is-invalid');
        inp_tarif.placeholder = 'Введіть тариф'
    }

```

```

        is_error = true;
    } else {
        inp_tarif.classList.remove('is-invalid');
        data.tarif = data.tarif;
    }

    if (data.sto.length < 3 || data.sto.length > 40)
{
        inp_sto.classList.add('is-invalid');
        inp_sto.placeholder = 'Введіть СТО'
        is_error = true;
    } else {
        inp_sto.classList.remove('is-invalid');
    }

    if (data.branch.length < 3 || data.branch.length
> 40) {
        inp_branch.classList.add('is-invalid');
        inp_branch.placeholder = 'Введіть відділ'
        is_error = true;
    } else {
        inp_branch.classList.remove('is-invalid');
    }

    if (data.position.length < 3 ||
data.position.length > 40) {
        inp_position.classList.add('is-invalid');
        inp_position.placeholder = 'Введіть посаду'
        is_error = true;
    } else {
        inp_position.classList.remove('is-invalid');
    }

    if (data.login.length < 3 || data.login.length >
20) {
        inp_login.classList.add('is-invalid');

```

```

        inp_login.placeholder = 'Введіть логін'
        is_error = true;
    } else {
        inp_login.classList.remove('is-invalid');
    }

    if (data.pwd != '' && (data.pwd.length < 3 ||
data.pwd.length > 20)) {
        inp_pwd.classList.add('is-invalid');
        inp_pwd.placeholder = 'Введіть пароль'
        is_error = true;
    } else {
        delete data.pwd;
        inp_pwd.classList.remove('is-invalid');
    }

    if (is_error) {
        document.getElementById('error').hidden =
false;

        document.getElementById('error').innerHTML =
'ERROR: Неправильно введені дані';
        return;
    }

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/employee/update/<?=$employee['employee_id'] ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status != 200) {
            document.getElementById('error').hidden
= false;

```

```

        document.getElementById('error').innerHT
ML = 'ERROR: ' + xhr.statusText;
        } else {
            document.getElementById('error').hidden
= true;
            document.getElementById('success').hidde
n = false;

            setTimeout(() => {
                window.location.href =
'/admin/employee/index';
            }, 5000);
        }
    };

    xhr.send(post_str(data));
});
})

function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

```

</html>

Файл app/admin/employee/view_index.php

<!DOCTYPE html>

<html data-bs-theme="dark">

<!--

License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

<head>

<meta charset="utf-8">

<meta name="author" content="Dmytro Zamkovyi">

<title><?= \$title ?></title>

<link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcdSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>


```

<?= $head ?>
<main class="container">
    <div style="margin-top: 16px;">
        <button id="new_employee" type="button" class="btn
btn-primary" data-bs-toggle="modal" data-bs-target="#add-
employee">Новий працівник</button>
        <output hidden id="success" class="text-
success">Користувача успішно додано!</output>

        <div class="modal fade" id="add-employee" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
                <div class="modal-content">
                    <div class="modal-header">
                        <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту працівника</h1>
                        <button type="button" class="btn-
close" data-bs-dismiss="modal" aria-label="Close"></button>
                    </div>
                    <div class="modal-body">
                        <div class="input-group mb-3">
                            <label class="input-group-text"
for="role">Роль</label>
                            <select class="form-select"
id="role">
                                <? foreach ($role_list as
$role) : ?>
                                    <? if ($role['active'])
: ?>
                                        <option value="<?=
$role['id'] ?>" selected><?= $role['role_name'] ?></option>
                                        <? else : ?>
                                            <option value="<?=
$role['id'] ?>"><?= $role['role_name'] ?></option>

```

```

        <? endif ?>
        <? endforeach ?>
    </select>
</div>

<div class="input-group mb-3">
    <span class="input-group-
text">ПІБ</span>

        <input id="first-name"
type="text" class="form-control" placeholder="Прізвище">
        <input id="last-name"
type="text" class="form-control" placeholder="Ім'я">
        <input id="father-name"
type="text" class="form-control" placeholder="Патронім">
    </div>

<div class="input-group mb-3">
    <span class="input-group-
text">Номер телефону</span>

        <input id="phone-number"
type="text" class="form-control">
        <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
    </div>

<div class="input-group mb-3">
    <span class="input-group-
text">Тариф</span>

        <input id="tarif" type="text"
class="form-control">

        <span class="input-group-
text">€</span>

        <span class="input-group-
text">0.00</span>
    </div>

```

```

        <div class="input-group mb-3">
            <span class="input-group-
text">CTO</span>

            <input id="sto" type="text"
class="form-control" list="sto-list">
            <datalist id="sto-list">
                <? foreach ($sto_list as
$sto) : ?>

                    <option value="<?=
$sto['sto_name'] ?>"></option>

                <? endforeach ?>
            </datalist>
        </div>

        <div class="input-group mb-3">
            <span class="input-group-
text">Біddіл</span>

            <input id="branch" type="text"
class="form-control" list="branch-list">
            <datalist id="branch-list">
                <? foreach ($branch_list as
$branch) : ?>

                    <option value="<%=
$branch['department_name'] ?>"></option>

                <? endforeach ?>
            </datalist>
        </div>

        <div class="input-group mb-3">
            <span class="input-group-
text">Посада</span>

            <input id="position" type="text"
class="form-control" list="position-list">
            <datalist id="position-list">
                <? foreach ($position_list
as $position) : ?>

```

```

<option value="<?=$position['position_name'] ?>"></option>
    <? endforeach ?>
</datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-
text">Логін</span>

    <input id="login" type="email"
class="form-control">

</div>

<div class="input-group mb-3">
    <span class="input-group-
text">Пароль</span>

    <input id="pwd" type="password"
class="form-control">

</div>
</div>
<div class="modal-footer">
    <output hidden id="error"
class="text-danger"></output>

    <button type="button" class="btn
btn-secondary" data-bs-dismiss="modal">Відміна</button>
    <button id="create" type="button"
class="btn btn-primary">Створити</button>

</div>
</div>
</div>
</div>

<hr>

<div class="accordion" id="according-search">

```

```

<div class="accordion-item">
    <h2 class="accordion-header">
        <button class="accordion-button collapsed"
type="button" data-bs-toggle="collapse" data-bs-
target="#collapse-search" aria-expanded="false" aria-
controls="collapseTwo">
            <h1>Пошук</h1>
        </button>
    </h2>
    <div id="collapse-search" class="accordion-
collapse collapse" data-bs-parent="#according-search">
        <div class="accordion-body">
            <div class="input-group mb-3">
                <label class="input-group-text"
for="search-role">Роль</label>
                <select class="form-select"
id="search-role">
                    <option value=""></option>
                    <? foreach ($role_list as $role)
: ?>
                        <option value="<?=
$role['id'] ?>"><?= $role['role_name'] ?></option>
                    <? endforeach ?>
                </select>
            </div>
            <div class="input-group mb-3">
                <span class="input-group-
text">ПІБ</span>
                <input id="search-first-name"
type="text" class="form-control" placeholder="Прізвище">
                <input id="search-last-name"
type="text" class="form-control" placeholder="Ім'я">
                <input id="search-father-name"
type="text" class="form-control" placeholder="Патронім">
            </div>

```

```

<div class="input-group mb-3">
    <span class="input-group-
text">CTO</span>

    <input id="search-sto" type="text"
class="form-control" list="search-sto-list">
    <datalist id="search-sto-list">
        <? foreach ($sto_list as $sto) :
?>

            <option value="<?=
$sto['sto_name'] ?>"></option>
        <? endforeach ?>
    </datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-
text">Відділ</span>

    <input id="search-branch"
type="text" class="form-control" list="search-branch-list">
    <datalist id="search-branch-list">
        <? foreach ($branch_list as
$branch) : ?>

            <option value="<?=
$branch['department_name'] ?>"></option>
        <? endforeach ?>
    </datalist>
</div>

<div class="input-group mb-3">
    <span class="input-group-
text">Посада</span>

    <input id="search-position"
type="text" class="form-control" list="search-position-list">
    <datalist id="search-position-list">

```

```

        <? foreach ($position_list as
$position) : ?>

            <option value="<?=$
$position['position_name'] ?>"></option>

        <? endforeach ?>
    </datalist>
</div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Логін</span>

        <input id="search-login"
type="email" class="form-control">
    </div>

    <div class="gap-2 d-md-flex justify-
content-md-end">

        <button id="search" type="button"
class="btn btn-primary">Пошук</button>

        <button id="search-reset"
type="button" class="btn btn-outline-primary">Скинути</button>
    </div>
</div>
</div>
</div>
<div style="margin-top: 16px;">
    <button id="reload" type="button" class="btn btn-
outline-success ms-auto">Оновити</button>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">ПІБ</th>
                <th scope="col">Роль</th>
                <th scope="col">Логін</th>

```

```

        <th scope="col">Телефон</th>
        <th scope="col">Місце роботи</th>
    </tr>
</thead>
<tbody id="table-body" class="table-group-
divider"></tbody>
</table>
</div>
</main>

<script>
    let employee = Array();
    window.addEventListener('DOMContentLoaded', () => {
        const modal = new
bootstrap.Modal(document.getElementById('add-employee'));
        const inp_role = document.getElementById('role');
        const inp_first_name =
document.getElementById('first-name');
        const inp_last_name = document.getElementById('last-
name');
        const inp_father_name =
document.getElementById('father-name');
        const inp_phone_number =
document.getElementById('phone-number');
        const inp_tarif = document.getElementById('tarif');
        const inp_sto = document.getElementById('sto');
        const inp_branch =
document.getElementById('branch');
        const inp_position =
document.getElementById('position');
        const inp_login = document.getElementById('login');
        const inp_pwd = document.getElementById('pwd');

        const btn_create =
document.getElementById('create');

```



```

        const btn_new_employee =
document.getElementById('new_employee');

        const inp_search_role =
document.getElementById('search-role');

        const inp_search_first_name =
document.getElementById('search-first-name');

        const inp_search_last_name =
document.getElementById('search-last-name');

        const inp_search_father_name =
document.getElementById('search-father-name');

        const inp_search_sto =
document.getElementById('search-sto');

        const inp_search_branch =
document.getElementById('search-branch');

        const inp_search_position =
document.getElementById('search-position');

        const inp_search_login =
document.getElementById('search-login');


        const btn_search =
document.getElementById('search');

        const btn_search_reset =
document.getElementById('search-reset');


        const btn_reload =
document.getElementById('reload');


        const table_body = document.getElementById('table-
body');


        send();


        /**
        * Функція для очищення полів пошуку
        */

```

```

btn_search_reset.addEventListener('click', () => {
    inp_search_role.value = '';
    inp_search_first_name.value = '';
    inp_search_last_name.value = '';
    inp_search_father_name.value = '';
    inp_search_sto.value = '';
    inp_search_branch.value = '';
    inp_search_position.value = '';
    inp_search_login.value = '';

    load_table();
});

btn_search.addEventListener('click', load_table);
btn_reload.addEventListener('click', send);

/**
 * Функція для очищення полів додавання нового
працівника
 */
btn_new_employee.addEventListener('click', () => {
    document.getElementById('success').hidden =
true;

    inp_first_name.value = '';
    inp_last_name.value = '';
    inp_father_name.value = '';
    inp_phone_number.value = '';
    inp_tarif.value = '';
    inp_sto.value = '';
    inp_branch.value = '';
    inp_position.value = '';
    inp_login.value = '';
    inp_pwd.value = '';
});

/**

```

```

* Функція для додавання нового працівника
*
* @returns void
*/
btn_create.addEventListener('click', () => {
    let is_error = false;
    let data = {
        role: inp_role.value,
        first_name: inp_first_name.value,
        last_name: inp_last_name.value,
        father_name: inp_father_name.value,
        phone_number: inp_phone_number.value,
        tarif: inp_tarif.value,
        sto: inp_sto.value,
        branch: inp_branch.value,
        position: inp_position.value,
        login: inp_login.value,
        pwd: inp_pwd.value
    };

    if (data.role == '') {
        inp_role.classList.add('is-invalid');
        inp_role.placeholder = 'Виберіть роль'
        is_error = true;
    } else {
        inp_role.classList.remove('is-invalid');
        data.role = data.role;
    }

    if (data.first_name.length < 3 ||
data.first_name.length > 40) {
        inp_first_name.classList.add('is-invalid');
        inp_first_name.placeholder = 'Введіть ім\'я'
        is_error = true;
    } else {

```

```

        inp_first_name.classList.remove('is-
invalid');

    }

    if (data.last_name.length < 3 ||
data.last_name.length > 40) {
        inp_last_name.classList.add('is-invalid');
        inp_last_name.placeholder = 'Введіть
прізвище'

        is_error = true;
    } else {
        inp_last_name.classList.remove('is-
invalid');
    }

    if (data.phone_number == '') {
        inp_phone_number.classList.add('is-
invalid');

        inp_phone_number.placeholder = 'Введіть
номер телефону'

        is_error = true;
    } else {
        inp_phone_number.classList.remove('is-
invalid');

        let re = /^[ ]*\+?([0-9]{2})[-\(\ ]*([0-
9]{1})[-\(\ ]*([0-9]{2})[-\(\ ]*([0-9]{3})[-\ ]*([0-9]{2})[-\ ]*([0-
9]{2})[ ]*$/gm;

        let res = re.exec(data.phone_number);
        if (res) {
            data.phone_number =
res.slice(1).join('');
        } else {
            inp_phone_number.classList.add('is-
invalid');

            inp_phone_number.placeholder = 'Невірний
формат номеру телефону'

```

```

        is_error = true;
    }
}

if (data.tarif == '') {
    inp_tarif.classList.add('is-invalid');
    inp_tarif.placeholder = 'Введіть тариф'
    is_error = true;
} else {
    inp_tarif.classList.remove('is-invalid');
    data.tarif = data.tarif;
}

if (data.sto.length < 3 || data.sto.length > 40)
{
    inp_sto.classList.add('is-invalid');
    inp_sto.placeholder = 'Введіть СТО'
    is_error = true;
} else {
    inp_sto.classList.remove('is-invalid');
}

if (data.branch.length < 3 || data.branch.length
> 40) {
    inp_branch.classList.add('is-invalid');
    inp_branch.placeholder = 'Введіть відділ'
    is_error = true;
} else {
    inp_branch.classList.remove('is-invalid');
}

if (data.position.length < 3 ||
data.position.length > 40) {
    inp_position.classList.add('is-invalid');
    inp_position.placeholder = 'Введіть посаду'
    is_error = true;
}

```

```

    } else {
        inp_position.classList.remove('is-invalid');
    }

    if (data.login.length < 3 || data.login.length >
20) {

        inp_login.classList.add('is-invalid');
        inp_login.placeholder = 'Введіть логін'
        is_error = true;
    } else {
        inp_login.classList.remove('is-invalid');
    }

    if (data.pwd.length < 3 || data.pwd.length > 20)
{

        inp_pwd.classList.add('is-invalid');
        inp_pwd.placeholder = 'Введіть пароль'
        is_error = true;
    } else {
        inp_pwd.classList.remove('is-invalid');
    }

    if (is_error) {
        document.getElementById('error').hidden =
false;

        document.getElementById('error').innerHTML =
'ERROR: Неправильно введені дані';
        return;
    }

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/employee/create');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

```

```

        xhr.onload = () => {
            if (xhr.status !== 200) {
                document.getElementById('error').hidden
= false;

                document.getElementById('error').innerHT
ML = 'ERROR: ' + xhr.statusText;
            } else {
                document.getElementById('error').hidden
= true;

                modal.hide();
                document.getElementById('success').hidde
n = false;

                setTimeout(() => {
                    document.getElementById('success').h
idden = true;

                    }, 10000);
                send();
            }
        };

        xhr.send(post_str(data));
    });

/**
 * Функція для завантаження таблиці
 *
 * @returns void
 */
function load_table() {
    let data = Object();

    if (inp_search_role.value !== '') {
        data['role'] = inp_search_role.value;
    }
}

```

```

        if (inp_search_first_name.value != '') {
            data['first_name'] =
inp_search_first_name.value;
        }

        if (inp_search_last_name.value != '') {
            data['last_name'] =
inp_search_last_name.value;
        }

        if (inp_search_father_name.value != '') {
            data['father_name'] =
inp_search_father_name.value;
        }

        if (inp_search_sto.value != '') {
            data['sto'] = inp_search_sto.value;
        }

        if (inp_search_branch.value != '') {
            data['branch'] = inp_search_branch.value;
        }

        if (inp_search_position.value != '') {
            data['position'] =
inp_search_position.value;
        }

        if (inp_search_login.value != '') {
            data['login'] = inp_search_login.value;
        }

        send(data);
    }

    /**

```



```

* Функція для відправки запиту
*
* @param {Array} inp_data - дані для відправки
* @returns void
*/
function send(inp_data = Array()) {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/employee/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            document.getElementById('table_body').in
nerHTML = '';
        } else {
            document.getElementById('error').hidden
= true;

            let result =
JSON.parse(xhr.responseText);
            employee = result;
            table_body.innerHTML = '';
            for (let i = 0; i < result.length; i++)
            {
                let tr =
document.createElement('tr');
                tr.classList.add('col-12', 'col-lg-
auto', 'my-2', 'justify-content-center', 'my-md-0');
                tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>'

                tr.innerHTML += '<td>' + sum_strs('
', result[i]['first_name'], result[i]['last_name'],
result[i]['fathers_name']) + '</td>'

                tr.innerHTML += '<td>' +
result[i]['role_name'] + '</td>'

```

```

        tr.innerHTML += '<td>' +
result[i]['login'] + '</td>'

        tr.innerHTML += '<td>+' +
result[i]['employee_pn'].slice(0, 3) + ' (' +
result[i]['employee_pn'].slice(3, 5) + ') ' +
result[i]['employee_pn'].slice(5, 8) + '-' +
result[i]['employee_pn'].slice(8, 10) + '-' +
result[i]['employee_pn'].slice(10) + '</td>'

        tr.innerHTML += '<td>' + sum_strs('
- ', result[i]['sto_name'], result[i]['department_name'],
result[i]['position_name']) + '</td>'

        tr.innerHTML += '<td>' + '<button
type="button" class="edit btn" style="padding: 0;"
onclick=edit(' + result[i]['employee_id'] + ')></button>'
+ '</td>'

        table_body.appendChild(tr);
    }
}

};

xhr.send(post_str(inp_data));
}

});

/**
 * Функція для перетворення об'єкту в строку
 *
 * @param {Object} data - об'єкт
 * @returns {String} - строка
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
}

```

```

        return str.slice(0, -1);
    }

    /**
     * Функція для склеювання строк
     *
     * @param {String} split - розділювач
     * @param {...String} theArgs - строки
     * @returns {String} - склеєна строка
     */
    function sum_strs(split, ...theArgs) {
        let total = '';
        for (const arg of theArgs) {
            if (arg == null) {
                continue;
            }
            total += arg + split;
        }
        return total.slice(0, -split.length);
    }

    /**
     * Функція для перенаправлення на сторінку редагування
    працівника
     *
     * @param {Number} i - id працівника
     * @returns void
     */
    function edit(i) {
        window.location.href = '/admin/employee/edit/' + i;
    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
```

```
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>
```

Файл app/admin/index/action.php

```
<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class adminIndex extends Curswork\ActionClass
{
    /**
     * Контроллер для перенаправления на головну сторінку
адміністратора
     *
     * @param $params
     * @param $data
     * @return mixed
     */
    public function action_index($params, $data)
```

```

        {
            $this->response->redirect('/admin/employee/index');
        }
    }
}

```

Файл `app/admin/order/action.php`

```
<?
```

```

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class adminOrder extends Curswork\ActionClass
{
    /**
     * Контроллер головної сторінки замовлень
     *
     * @param string $params - параметри
     * @param array $data - дані
     * @return void
     */
    public function action_index($params, $data)
    {

```

```

$view = "index";
$data['label'] = 'Перегляд замовлень';
$data['menu'] = array(
    array(
        'name' => 'Працівники',
        'href' => '/admin/employee/index',
        'active' => false,
        'img' => '/data/icon/employee.png'
    ),
    array(
        'name' => 'Послуги',
        'href' => '/admin/service/index',
        'active' => false,
        'img' => '/data/icon/service.png'
    ),
    array(
        'name' => 'Замовлення',
        'href' => '/admin/order/index',
        'active' => true,
        'img' => '/data/icon/order_activ.png'
    ),
    array(
        'name' => 'Клієнти',
        'href' => '/admin/client/index',
        'active' => false,
        'img' => '/data/icon/client.png'
    ),
    array(
        'name' => 'Вихід',
        'href' => '/index/index/logout',
        'active' => false,
        'img' => '/data/icon/logout.png'
    ),
);
$data['head'] = $this->load('index', 'index', 'head',
'', $data);

```

```

        $data['title'] = 'Замовлення · ' . $this->config->projectName;

        $data['success'] = $this->session->success == null ?
'hidden' : '';
        unset($this->session->success);

        $this->view->out($view, $data);
    }

    /**
     * Контроллер для пошуку замовлень
     *
     * @param string $params - параметри
     * @param array $data - дані
     * @return void
     */
    public function action_search($params, $data)
    {
        $data['first_name'] = isset($this->request->post->first_name) ? urldecode($this->request->post->first_name) :
null;

        $data['last_name'] = isset($this->request->post->last_name) ? urldecode($this->request->post->last_name) : null;

        $data['father_name'] = isset($this->request->post->father_name) ? urldecode($this->request->post->father_name) :
null;

        $data['phone_number'] = isset($this->request->post->phone_number) ? urldecode($this->request->post->phone_number) :
null;

        $data['car'] = isset($this->request->post->car) ?
urldecode($this->request->post->car) : null;

        $data['start_date'] = isset($this->request->post->start_date) ? urldecode($this->request->post->start_date) :
null;

        $data['end_date'] = isset($this->request->post->end_date) ? urldecode($this->request->post->end_date) : null;
    }

```

```

        $data['complete'] = isset($this->request->post->complete) ? urldecode($this->request->post->complete) : null;
        $data['pay'] = isset($this->request->post->pay) ? urldecode($this->request->post->pay) : null;

        if ($data['first_name'] == null) {
            unset($data['first_name']);
        } elseif (strlen($data['first_name']) < 3 || strlen($data['first_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['last_name'] == null) {
            unset($data['last_name']);
        } elseif (strlen($data['last_name']) < 3 || strlen($data['last_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['father_name'] == null) {
            unset($data['father_name']);
        } elseif (strlen($data['father_name']) < 3 || strlen($data['father_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['phone_number'] == null) {
            unset($data['phone_number']);
        } else {
            preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/', $data['phone_number'], $matches);
            if (count($matches[0]) == 0) {

```



```

        header('HTTP/1.1 400 Bad Request');
        return;
    }
    $matches = array_slice($matches, 1);
    for ($i = 0; $i < count($matches); $i++) {
        $matches[$i] = (int)$matches[$i][0];
    }
    $data['phone_number'] = (int)implode($matches);
}

if ($data['car'] == null) {
    unset($data['car']);
} elseif (strlen($data['car']) == 0 ||
strlen($data['car']) > 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if ($data['start_date'] == null) {
    unset($data['start_date']);
} elseif (strlen($data['start_date']) != 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if ($data['end_date'] == null) {
    unset($data['end_date']);
} elseif (strlen($data['end_date']) != 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if (isset($data['start_date']) &&
isset($data['end_date'])) {
    if (strtotime($data['start_date']) >
strtotime($data['end_date'])) {

```

```

        header('HTTP/1.1 400 Bad Request');
        return;
    }
}

if ($data['complete'] == null) {
    unset($data['complete']);
} elseif ($data['complete'] != 'true' &&
$data['complete'] != 'false') {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if ($data['pay'] == null) {
    unset($data['pay']);
} elseif ($data['pay'] != 'true' && $data['pay'] !=
'false') {
    header('HTTP/1.1 400 Bad Request');
    return;
}

echo json_encode($this->model->search($data));
}

/**
 * Контроллер для редагування замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_update($params, $data)
{
    $data['order_id'] = $params;
    $data['discount'] = (int)urldecode($this->request->post-
>discount);

```

```

        if (!$this->model->update($data)) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        header('HTTP/1.1 200 Ok');
    }

/**
 * Контроллер для видалення замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_delete($params, $data)
{
    $data['order_id'] = (int)$params;
    $this->model->delete($data['order_id']);
    $this->response->redirect('/admin/order/index');
}

/**
 * Контроллер для створення замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_add($params, $data)
{
    $this->session->success = true;
}

/**
 * Контроллер для перегляду сторінки створення замовлення

```

```

*
* @param string $params - параметри
* @param array $data - дані
* @return void
*/
public function action_create($params, $data)
{
    $view = "create";

    $data['title'] = 'Створення замовлення · ' . $this->config->projectName;

    $data['sto_list'] = $this->model->get_sto_list();

    $this->view->out($view, $data);
}

/**
 * Контроллер для перегляду сторінки редагування замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_edit($params, $data)
{
    $data['order_id'] = $params;
    $view = "edit";

    $data['order_info'] = $this->model->get_order($data['order_id']);
    if (!$data['order_info']) {
        header('HTTP/1.1 404 Not Found');
        $this->response->redirect('/index/index/404');
        return;
    }
}

```

```

        $data['task_list'] = $this->model-
>get_task($data['order_id']);
        if (!$data['task_list']) {
            header('HTTP/1.1 404 Not Found');
            $this->response->redirect('/index/index/404');
            return;
        }

        $data['sto_info'] = $this->model-
>get_sto($data['task_list'][0]['employee_id']);

        $data['title'] = 'Редагування замовлення · ' . $this-
>config->projectName;

        $this->view->out($view, $data);
    }

/**
 * Контроллер для отримання списку завдань замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_task($params, $data)
{
    $data['order_id'] = $params;

    $data['task_list'] = $this->model-
>get_task($data['order_id']);
    if (!$data['task_list']) {
        header('HTTP/1.1 404 Not Found');
        $this->response->redirect('/index/index/404');
        return;
    }
}

```

```

        echo json_encode($data['task_list']);
    }

/**
 * Контроллер для видалення завдання замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_delete_task($params, $data)
{
    $data['task_id'] = $params;

    if (!$this->model->delete_task($data['task_id'])) {
        header('HTTP/1.1 400 Bad Request');
    } else {
        header('HTTP/1.1 200 Ok');
    }
}

/**
 * Контроллер для отримання списку працівників СТО
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_employee_list($params, $data)
{
    $data['sto_name'] = urldecode($this->request->post->sto_name);

    $result = $this->model->get_employee_list($data['sto_name']);

```

```

        if (!$result) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        echo json_encode($result);
    }

/**
 * Контроллер для отримання списку послуг
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_task_list($params, $data)
{
    $result = $this->model->get_task_list();
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

/**
 * Контроллер для додавання завдання замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_add_task($params, $data)
{
    $data['order_id'] = urldecode($params);
    $data['name'] = explode(' ', urldecode($this->request-
>post->name));

```

```

        $data['task'] = urldecode($this->request->post->task);

        if (!$this->model->add_task($data)) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        header('HTTP/1.1 200 Ok');
    }

/**
 * Контроллер для пошуку клієнтів
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з клієнтами
 */
public function action_search_client($params, $data)
{
    $data['search'] = urldecode($this->request->post->search);

    $result = $this->model->search_client($data['search']);
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

/**
 * Контроллер для додавання клієнта
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з клієнтом
 */

```



```

public function action_add_client($params, $data)
{
    $data['first_name'] = urldecode($this->request->post->first_name);
    $data['last_name'] = urldecode($this->request->post->last_name);
    $data['fathers_name'] = $this->request->post->father_name;
    $data['phone_number'] = urldecode($this->request->post->phone_number);

    $search = $this->model->search_client_by_phone($data['phone_number']);

    if ($search) {
        header('HTTP/1.1 200 Ok');
        echo json_encode($search);
        return;
    }

    if (strlen($data['first_name']) < 3 ||
        strlen($data['first_name']) > 40) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if (strlen($data['last_name']) < 3 ||
        strlen($data['last_name']) > 40) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/', $data['phone_number'], $matches);
    if (count($matches[0]) == 0) {

```

```

        header('HTTP/1.1 400 Bad Request');
        return;
    }
    $matches = array_slice($matches, 1);
    for ($i = 0; $i < count($matches); $i++) {
        $matches[$i] = (int)$matches[$i][0];
    }
    $data['phone_number'] = (int)implode($matches);

    $add = $this->model->add_client($data);
    if (!$add) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($this->model->
>search_client_by_phone($data['phone_number']));
    }

/**
 * Контроллер для пошуку автомобілів
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з автомобілями
 */
public function action_search_car($params, $data)
{
    $data['search'] = urldecode($this->request->post-
>search);

    $result = $this->model->search_car($data['search']);
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

```

```

}

/**
 * Контроллер для додавання автомобіля
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з автомобілем
 */
public function action_add_car($params, $data)
{
    $data['make'] = urldecode($this->request->post->make);
    $data['model'] = urldecode($this->request->post->model);
    $data['year'] = urldecode($this->request->post-
>car_year);
    $data['license_plate'] = urldecode($this->request->post-
>license_plate);
    $data['description'] = urldecode($this->request->post-
>description);

    $search = $this->model-
>search_car($data['license_plate']);

    if ($search) {
        header('HTTP/1.1 200 Ok');
        echo json_encode($search);
        return;
    }

    if (strlen($data['make']) < 3 || strlen($data['make']) >
40) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
}

```

```

        if (strlen($data['model']) < 3 || strlen($data['model'])
> 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if (strlen($data['year']) != 4) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if (strlen($data['license_plate']) < 3 ||
strlen($data['license_plate']) > 10) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        $add = $this->model->add_car($data);

        if (!$add) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        echo json_encode($this->model->search_car($data['license_plate']));
    }

    /**
     * Контроллер для створення замовлення
     *
     * @param string $params - параметри
     * @param array $data - дані
     * @return void - масив з замовленням
     */
    public function action_create_order($params, $data)
    {

```

```

        $data['client_id'] = urldecode($this->request->post->client_id);
        $data['car_id'] = urldecode($this->request->post->car_id);

        $result = $this->model->create_order($data);
        if (!$result) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        echo json_encode($result);
    }
}

```

Файл app/admin/order/model.php

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class adminOrderModel extends Curswork\ActionClassModel
{

```

```

/**
 * Метод моделі для отримання списку замовлень
 *
 * @param $data - масив з даними для пошуку
 * @return array - масив з замовленнями
 */
public function search($data)
{
    $sql = 'SELECT "order".id, client.first_name,
client.last_name, client.fathers_name, client.phone_number,
car.make, car.model, car.license_plate, "order".create_date,
"order".completed, "order".payment FROM "order", client, car
WHERE "order".id_client = client.id AND "order".id_car =
car.id';

    if (isset($data['first_name'])) {
        $sql .= ' AND client.first_name LIKE \'%' . $this-
>db->escape($data['first_name']) . '%\'';
    }
    if (isset($data['last_name'])) {
        $sql .= ' AND client.last_name LIKE \'%' . $this-
>db->escape($data['last_name']) . '%\'';
    }
    if (isset($data['father_name'])) {
        $sql .= ' AND client.fathers_name LIKE \'%' . $this-
>db->escape($data['father_name']) . '%\'';
    }
    if (isset($data['phone_number'])) {
        $sql .= ' AND client.phone_number = ' . $this->db-
>escape($data['phone_number']);
    }
    if (isset($data['car'])) {
        $sql .= ' AND car.license_plate LIKE \'%' . $this-
>db->escape($data['car']) . '%\'';
    }
}

```

```

        if (isset($data['start_date'])) {
            $sql .= ' AND "order".create_date >= \'' . $this->db->escape($data['start_date']) . '\'';
        }
        if (isset($data['end_date'])) {
            $sql .= ' AND "order".create_date <= \'' . $this->db->escape($data['end_date']) . '\'';
        }
        if (isset($data['complete'])) {
            $sql .= ' AND "order".completed = ' . $this->db->escape($data['complete']);
        }
        if (isset($data['pay'])) {
            $sql .= ' AND "order".payment = ' . $this->db->escape($data['pay']);
        }

        $sql .= ';';

        $result = $this->db->query($sql);

        if ($this->db->last_error) {
            return array();
        }

        for ($i = 0; $i < count($result); $i++) {
            $result[$i]['completed'] = $result[$i]['completed']
            == 't';
            $result[$i]['payment'] = $result[$i]['payment'] ==
            't';
        }
        return $result;
    }

    /**
     * Метод моделі для отримання деталей замовлення

```

```

*
* @param $id - id замовлення
* @return bool|array - масив з деталями замовлення
*/
public function get_order($id)
{
    $sql = 'SELECT "order".create_date, "order".completed,
"order".payment, "order".discount AS order_discount, car.make,
car.model, car.license_plate, client.first_name,
client.last_name, client.fathers_name, client.phone_number,
client.discount FROM "order", car, client WHERE "order".id_car =
car.id AND "order".id_client = client.id AND "order".id = ' .
$this->db->escape($id) . ' ';
    $order_result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($order_result) == 0) {
        return false;
    }
    return $order_result[0];
}

/**
* Метод моделі для отримання списку завдань по замовленню
*
* @param $id - id замовлення
* @return bool|array - масив з завданнями
*/
public function get_task($id)
{
    $sql = 'SELECT task.id, task.is_done, employee.id AS
employee_id, employee.first_name, employee.last_name,
employee.fathers_name, service.service_name, service.price FROM
task, employee, service WHERE task.id_employee = employee.id AND

```



```

task.id_service = service.id AND task.id_order = ' . $this->db-
>escape($id) . ' ';
    $tasks_result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return $tasks_result;
}

/**
 * Метод моделі для отримання списку СТО
 *
 * @param $employee_id - id працівника
 * @return bool|array - масив з СТО
 */
public function get_sto($employee_id)
{
    $sql = 'SELECT sto.id, sto.sto_name FROM employee, sto
WHERE employee.id_sto = sto.id AND employee.id = ' . $this->db-
>escape($employee_id) . ' ';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return $result[0];
}

/**
 * Метод моделі для видалення завдання по замовленню
 *
 * @param $task_id - id завдання
 * @return bool - результат видалення
 */
public function delete_task($task_id)
{

```

```

        $sql = 'DELETE FROM task WHERE id = ' . $this->db->escape($task_id) . ' ';
        $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return true;
    }

    /**
     * Метод моделі для отримання списку працівників СТО
     *
     * @param $sto_name - назва СТО
     * @return bool|array - масив з працівниками
     */
    public function get_employee_list($sto_name)
    {
        $sql = 'SELECT employee.id, employee.first_name,
employee.last_name, employee.fathers_name FROM employee, sto
WHERE employee.id_sto = sto.id AND employee.id_role = 3 AND
sto.sto_name = \'' . $this->db->escape($sto_name) . '\';';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return $result;
    }

    /**
     * Метод моделі для отримання списку послуг
     *
     * @return bool|array - масив з послугами
     */
    public function get_task_list()
    {
        $sql = 'SELECT service.service_name FROM service;';

```

```

        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return $result;
    }

    /**
     * Метод моделі для додавання завдання по замовленню
     *
     * @param $data - масив з даними для додавання
     * @return bool - результат додавання
     */
    public function add_task($data)
    {
        $sql = 'SELECT id FROM employee WHERE first_name = \'' .
$this->db->escape($data['name'][0]) . '\';
        if (isset($data['name'][1])) {
            $sql .= ' AND last_name = \'' . $this->db->
>escape($data['name'][1]) . '\';
        }
        if (isset($data['name'][2])) {
            $sql .= ' AND fathers_name = \'' . $this->db->
>escape($data['name'][2]) . '\';
        }
        $id_employee = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        if (count($id_employee) == 0) {
            return false;
        }
        $id_employee = $id_employee[0]['id'];

        $sql = 'SELECT id FROM service WHERE service_name = \''
. $this->db->escape($data['task']) . '\';';
    }

```

```

        $id_service = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        if (count($id_service) == 0) {
            return false;
        }
        $id_service = $id_service[0]['id'];

        $sql = 'INSERT INTO task (id_order, id_service,
id_employee, is_done) VALUES ('. $this->db-
>escape($data['order_id']) . ', ' . $id_service . ', '.
$id_employee . ', false)';
        $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return true;
    }

/**
 * Метод моделі для зміни деталей замовлення
 *
 * @param $data - масив з даними для зміни
 * @return bool - результат зміни
 */
public function update($data)
{
    if ($data['discount'] < 0 || $data['discount'] > 100) {
        return false;
    }

    if ($data['discount'] == 0) {
        $data['discount'] = 'NULL';
    }
}

```

```

        $sql = 'UPDATE "order" SET discount = '. $this->db-
>escape($data['discount']) . ' WHERE id = '. $this->db-
>escape($data['order_id']) .';';

        $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return true;
    }

    /**
     * Метод моделі для втдалення замовлення
     *
     * @param $id - id замовлення
     * @return void
     */
    public function delete($id)
    {

        $sql = 'DELETE FROM task WHERE id_order = ' . $this->db-
>escape($id) . ';';
        $this->db->query($sql);

        $sql = 'DELETE FROM "order" WHERE id = ' . $this->db-
>escape($id) . ';';
        $this->db->query($sql);
    }

    /**
     * Метод моделі для отримання списку СТО
     *
     * @return bool|array - масив з СТО
     */
    public function get_sto_list()
    {

```

```

        $sql = 'SELECT id, sto_name FROM sto;';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return $result;
    }

    /**
     * Метод моделі для пошуку клієнта
     *
     * @return bool|array - масив з відділеннями
     */
    public function search_client($search)
    {
        $sql = 'SELECT * FROM client WHERE first_name LIKE \'%'.
        $this->db->escape($search) . '%\' OR last_name LIKE \'%'. $this-
        >db->escape($search) . '%\' OR fathers_name LIKE \'%'. $this-
        >db->escape($search) . '%\';';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        if (count($result) != 1) {
            return false;
        }
        return $result;
    }

    /**
     * Метод моделі для пошуку клієнта по номеру телефону
     *
     * @return bool|array - масив з відділеннями
     */
    public function search_client_by_phone($phone)
    {

```

```

        $sql = 'SELECT * FROM client WHERE phone_number = ' .
$this->db->escape($phone) . ' ';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        if (count($result) != 1) {
            return false;
        }
        return $result;
    }

/**
 * Метод моделі для додавання клієнта
 *
 * @param $data - масив з даними для додавання
 * @return bool - результат додавання
 */
public function add_client($data)
{
    if (!isset($data['fathers_name'])) {
        $sql = 'INSERT INTO client (first_name, last_name,
phone_number) VALUES (\'' . $this->db-
>escape($data['first_name']) . '\', \'' . $this->db-
>escape($data['last_name']) . '\', ' . $this->db-
>escape($data['phone_number']) . ');';
    } else {
        $sql = 'INSERT INTO client (first_name, last_name,
fathers_name, phone_number) VALUES (\'' . $this->db-
>escape($data['first_name']) . '\', \'' . $this->db-
>escape($data['last_name']) . '\', \'' . $this->db-
>escape((string)$data['fathers_name']) . '\', ' . $this->db-
>escape($data['phone_number']) . ');';
    }
    $this->db->query($sql);
    if ($this->db->last_error) {

```

```

        return false;
    }
    return true;
}

/**
 * Метод моделі для пошуку автомобіля
 *
 * @return bool|array - масив з відділеннями
 */
public function search_car($search)
{
    $sql = 'SELECT * FROM car WHERE license_plate = \'' .
$this->db->escape($search) . '\';';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($result) != 1) {
        return false;
    }
    return $result;
}

/**
 * Метод моделі для додавання автомобіля
 *
 * @param $data - масив з даними для додавання
 * @return bool - результат додавання
 */
public function add_car($data)
{
    if (!isset($data['year']) &&
!isset($data['description'])) {
        $sql = 'INSERT INTO car (make, model, license_plate)
VALUES (\'' . $this->db->escape($data['make']) . '\', \'' .

```



```

$this->db->escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\)';
    } else {
        if (!isset($data['year'])) {
            $sql = 'INSERT INTO car (make, model,
license_plate) VALUES (\'' . $this->db->escape($data['make']) .
'\', \'' . $this->db->escape($data['model']) . '\', \'' . $this-
>db->escape($data['license_plate']) . '\)';
        } else {
            $sql = 'INSERT INTO car (make, model,
license_plate, car_year) VALUES (\'' . $this->db-
>escape($data['make']) . '\', \'' . $this->db-
>escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\', ' . $this->db-
>escape($data['year']) . ')';
        }

        if (!isset($data['description'])) {
            $sql = 'INSERT INTO car (make, model,
license_plate) VALUES (\'' . $this->db->escape($data['make']) .
'\', \'' . $this->db->escape($data['model']) . '\', \'' . $this-
>db->escape($data['license_plate']) . '\)';
        } else {
            $sql = 'INSERT INTO car (make, model,
license_plate, description) VALUES (\'' . $this->db-
>escape($data['make']) . '\', \'' . $this->db-
>escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\', \'' . $this->db-
>escape($data['description']) . '\)';
        }
    }

    if (isset($data['year']) && isset($data['description']))
    {
        $sql = 'INSERT INTO car (make, model, license_plate,
car_year, description) VALUES (\'' . $this->db-

```

```

>escape($data['make']) . '\', \' . $this->db-
>escape($data['model']) . '\', \' . $this->db-
>escape($data['license_plate']) . '\', ' . $this->db-
>escape($data['year']) . ', \' . $this->db-
>escape($data['description']) . '\)';
    }

    $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return true;
}

/**
 * Метод моделі для створення замовлення
 *
 * @param $data - масив з даними для створення
 * @return bool - результат створення
 */
public function create_order($data)
{
    $sql = 'INSERT INTO "order" (id_client, id_car,
create_date, completed, payment) VALUES (' . $this->db-
>escape($data['client_id']) . ', ' . $this->db-
>escape($data['car_id']) . ', now(), false, false) RETURNING
id;';

    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return $result;
}
}

```

Файл `app/admin/order/view_create.php`

```
<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
    <header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">
```

```

        <a id="back" href="/admin/order/index" class="d-flex
align-items-center mb-3 mb-md-0 me-md-auto link-body-emphasis
text-decoration-none" style="margin-left: 16px;">
            
            <span class="fs-4">Назад</span>
        </a>

        <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;" hidden>Зберегти</button>
    </header>
    <main>
        <div class="container">
            <div class="input-group mb-3">
                <span class="input-group-text">Замовник</span>
                <input id="client" type="text" class="form-
control" disabled>
                <button id="add-client-btn" class="btn btn-
outline-primary" type="button" data-bs-toggle="modal" data-bs-
target="#add-client-model">Вибрати</button>
            </div>

            <div id="car-div" class="input-group mb-3" hidden>
                <span class="input-group-text">Автомобіль</span>
                <input id="car" type="text" class="form-control"
disabled>
                <button id="add-car-btn" class="btn btn-outline-
primary" type="button" data-bs-toggle="modal" data-bs-
target="#add-car-model">Вибрати</button>
            </div>

            <div id="sto-list-div" class="input-group mb-3"
hidden>
                <span class="input-group-text">СТО</span>
                <select id="sto-list" class="form-select"
autocomplete="off">

```

```

        <option selected hidden value=""></option>
        <? foreach ($sto_list as $item) : ?>
            <option value="<?= $item['sto_name']
?>"><?= $item['sto_name'] ?></option>
        <? endforeach; ?>
    </select>
</div>

    <p id="error" class="text-danger" hidden></p>
</div>

    <div id="new-task-div" class="container" hidden>
        <button id="new-task" type="button" class="btn btn-
primary" data-bs-toggle="modal" data-bs-target="#add-task-
model">Нове завдання</button>
    </div>

    <div id="table-div" style="margin-left: 16px; margin-
right: 16px;" hidden>
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">#</th>
                    <th scope="col">Назва послуги</th>
                    <th scope="col">Виконавець</th>
                    <th scope="col">Ціна</th>
                    <th scope="col">Виконано</th>
                    <th scope="col"></th>
                </tr>
            </thead>
            <tbody id="table-body"></tbody>
        </table>
    </div>

```

```

        <div class="modal fade" id="add-client-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
                <div class="modal-content">
                    <div class="modal-header">
                        <h1 class="modal-title fs-5"
id="staticBackdropLabel">Додавання клієнта</h1>
                        <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
                    </div>
                    <div class="modal-body">
                        <div class="input-group mb-3">
                            <input id="client-search"
type="text" class="form-control" placeholder="Пошук">
                            <button id="client-search-btn"
class="btn btn-outline-primary" type="button">Знайти</button>
                        </div>

                        <div class="input-group mb-3">
                            <span class="input-group-
text">ПІБ</span>
                            <input id="first-name" type="text"
class="form-control" placeholder="Прізвище">
                            <input id="last-name" type="text"
class="form-control" placeholder="Ім'я">
                            <input id="fathers-name" type="text"
class="form-control" placeholder="Патронім">
                        </div>

                        <div class="input-group mb-3">
                            <span class="input-group-text">Номер
телефону</span>
                            <input id="phone-number" type="text"
class="form-control">

```

```

        <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
    </div>
</div>
<div class="modal-footer">
    <output hidden id="client-error"
class="text-danger"></output>
    <button id="add-client" type="button"
class="btn btn-primary">Додати</button>
</div>
</div>
</div>
</div>

<div class="modal fade" id="add-car-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="staticBackdropLabel">Додавання автомобіля</h1>
                <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="input-group mb-3">
                    <input id="car-search" type="text"
class="form-control" placeholder="Пошук">
                    <button id="car-search-btn"
class="btn btn-outline-primary" type="button">Знайти</button>
                </div>

                <div class="input-group mb-3">

```

```

        <span class="input-group-
text">Мапка</span>

        <input id="make" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Модель</span>

        <input id="model" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Рік
випуску</span>

        <input id="year" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Номерний знак</span>

        <input id="license-plate"
type="text" class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Опис
(опціонально)</span>

        <textarea id="description"
class="form-control" aria-label="With textarea"></textarea>
    </div>
</div>
<div class="modal-footer">

```



```

        <output hidden id="car-error"
class="text-danger"></output>
        <button id="add-car" type="button"
class="btn btn-primary">Додати</button>
    </div>
</div>
</div>
</div>

<div class="modal fade" id="add-task-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту клієнта</h1>
                <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="input-group mb-3">
                    <span class="input-group-
text">Працівник</span>
                    <input id="employee" type="text"
class="form-control" list="employee-list">
                    <datalist id="employee-
list"></datalist>
                </div>

                <div class="input-group mb-3">
                    <span class="input-group-
text">Завдання</span>
                    <input id="task" type="text"
class="form-control" list="task-list">

```

```

        <datalist id="task-list"></datalist>
    </div>
</div>
<div class="modal-footer">
    <output hidden id="error" class="text-
danger"></output>
    <button id="create" type="button"
class="btn btn-primary">Створити</button>
</div>
</div>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const task_modal = new
bootstrap.Modal(document.getElementById('add-task-model'));
        const client_modal = new
bootstrap.Modal(document.getElementById('add-client-model'));
        const car_modal = new
bootstrap.Modal(document.getElementById('add-car-model'));

        const inp_client =
document.getElementById('client');
        const inp_car = document.getElementById('car');
        const inp_sto_list = document.getElementById('sto-
list');

        const error = document.getElementById('error');
        const btn_new_task = document.getElementById('new-
task');

        const table_div = document.getElementById('table-
div');

        const new_task_div = document.getElementById('new-
task-div');

```

```

        const car_div = document.getElementById('car-div');
        const sto_list_div = document.getElementById('sto-
list-div');

        const inp_client_search =
document.getElementById('client-search');
        const btn_client_search =
document.getElementById('client-search-btn');
        const client_error =
document.getElementById('client-error');
        const btn_add_client = document.getElementById('add-
client');
        const inp_client_first_name =
document.getElementById('first-name');
        const inp_client_last_name =
document.getElementById('last-name');
        const inp_client_fathers_name =
document.getElementById('fathers-name');
        const inp_client_phone_number =
document.getElementById('phone-number');

        const inp_car_search = document.getElementById('car-
search');
        const btn_car_search = document.getElementById('car-
search-btn');
        const car_error = document.getElementById('car-
error');
        const btn_add_car = document.getElementById('add-
car');
        const inp_car_make =
document.getElementById('make');
        const inp_car_model =
document.getElementById('model');
        const inp_car_year =
document.getElementById('year');

```

```

        const inp_car_license_plate =
document.getElementById('license-plate');
        const inp_car_description =
document.getElementById('description');

        const btn_client = document.getElementById('add-
client-btn');
        const btn_car = document.getElementById('add-car-
btn');

        const employee_list =
document.getElementById('employee-list');
        const task_list = document.getElementById('task-
list');

        const btn_create =
document.getElementById('create');
        const inp_employee =
document.getElementById('employee')
        const inp_task = document.getElementById('task');

        const btn_save = document.getElementById('save');

        let client_id = null;
        let car_id = null;
        let order_id = null;

        inp_client.value = '';
        inp_car.value = '';
        task_list.value = '';

        /**
         * Функція для очищення полів
         *
         * @param void
         */

```

```

    btn_client.addEventListener('click', () => {
        inp_client_search.value = '';
        inp_client_first_name.value = '';
        inp_client_last_name.value = '';
        inp_client_fathers_name.value = '';
        inp_client_phone_number.value = '';
    });

/**
 * Функція для пошуку клієнта
 *
 * @param void
 */
    btn_client_search.addEventListener('click', () => {
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/admin/order/search_client');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                client_error.hidden = true;
                client_id = data[0].id;
                inp_client_first_name.value =
data[0].first_name;
                inp_client_last_name.value =
data[0].last_name;
                inp_client_fathers_name.value =
data[0].fathers_name;
                inp_client_phone_number.value =
data[0].phone_number;
            } else {
                client_error.hidden = false;
            }
        }
    });

```

```

        client_error.innerText = 'Клієнт не
знайдений';

        setTimeout(() => {
            client_error.hidden = true;
        }, 3000);
    }
};

xhr.send(post_str({
    'search': inp_client_search.value
}));
});

/**
 * Функція для додавання клієнта
 *
 * @param void
 */
btn_add_client.addEventListener('click', () => {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/add_client');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            client_modal.hide();
            let data = JSON.parse(xhr.response);
            inp_client.value = sum_strs(' ',
data[0].first_name, data[0].last_name, data[0].fathers_name);
            client_id = data[0].id;
            car_div.hidden = false;
        } else {
            client_error.hidden = false;
            client_error.innerText = 'Клієнт не
доданий';

```

```

        setTimeout(() => {
            client_error.hidden = true;
        }, 3000);
    }
};

xhr.send(post_str({
    'first_name': inp_client_first_name.value,
    'last_name': inp_client_last_name.value,
    'fathers_name':
inp_client_fathers_name.value,
    'phone_number':
inp_client_phone_number.value
}));

});

// TODO: Додати перевірку на введення даних

/**
 * Функція для очищення полів
 *
 * @param void
 */
btn_car.addEventListener('click', () => {
    inp_car_search.value = '';
    inp_car_make.value = '';
    inp_car_model.value = '';
    inp_car_year.value = '';
    inp_car_license_plate.value = '';
    inp_car_description.value = '';
});

/**
 * Функція для пошуку автомобіля
 *
 * @param void
 */
btn_car_search.addEventListener('click', () => {

```

```

        console.log(111);
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/admin/order/search_car');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                car_error.hidden = true;
                inp_car_make.value = data[0].make;
                inp_car_model.value = data[0].model;
                inp_car_year.value = data[0].car_year;
                inp_car_license_plate.value =
data[0].license_plate;
                inp_car_description.value =
data[0].description;
            } else {
                car_error.hidden = false;
                car_error.innerText = 'Автомобіль не
знайдений';

                setTimeout(() => {
                    car_error.hidden = true;
                }, 3000);
            }
        };
        xhr.send(post_str({
            'search': inp_car_search.value
        }));
    });

/**
 * Функція для додавання автомобіля
 *
 * @param void

```



```

        */
        btn_add_car.addEventListener('click', () => {
            let xhr = new XMLHttpRequest();
            xhr.open('POST', '/admin/order/add_car');
            xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
            xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
            xhr.onload = () => {
                if (xhr.status == 200) {
                    car_modal.hide();
                    let data = JSON.parse(xhr.response);
                    inp_car.value = sum_strs(' ',
data[0].make, data[0].model, data[0].car_year);
                    car_id = data[0].id;
                    sto_list_div.hidden = false;
                    inp_sto_list.disabled = false;
                    create_order();
                } else {
                    car_error.hidden = false;
                    car_error.innerText = 'Автомобіль не
доданий';

                    setTimeout(() => {
                        car_error.hidden = true;
                    }, 3000);
                }
            };
            xhr.send(post_str({
                'make': inp_car_make.value,
                'model': inp_car_model.value,
                'car_year': inp_car_year.value,
                'license_plate':
inp_car_license_plate.value,
                'description': inp_car_description.value
            }));
        });
    });

```

```

/**
 * Функція для вибору СТО
 *
 * @param void
 */
inp_sto_list.addEventListener('change', () => {
    table_div.hidden = false;
    new_task_div.hidden = false;
    btn_new_task.disabled = false;
    get_task_list();
    get_employee_list();
});

/**
 * Функція для додавання завдання
 *
 * @param void
 */
btn_create.addEventListener('click', () => {
    let data = Object();
    data['name'] = inp_employee.value;
    data['task'] = inp_task.value;

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/add_task/' +
order_id);

    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;

```

```

        } else {
            error.hidden = true;
            task_modal.hide();
            get_task();
            document.getElementById('save').hidden =
false;

        }
    };

    xhr.send(post_str(data));
});

/**
 * Функція для очищення полів
 *
 * @param void
 */
btn_new_task.addEventListener('click', () => {
    inp_employee.value = '';
    inp_task.value = '';
});

/**
 * Функція для збереження замовлення
 *
 * @param void
 */
btn_save.addEventListener('click', () => {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/add');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.send();
    window.location.href = '/admin/order/index'

```

```

    });

    /**
     * Функція для отримання списку працівників
     *
     * @param void
     */
    function get_employee_list() {
        let xhr = new XMLHttpRequest();
        xhr.open('POST',
'/admin/order/get_employee_list');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                employee_list.innerHTML = '';
                data.forEach((item) => {
                    let option =
document.createElement('option');
                    option.value = sum_strs(' ',
item.first_name, item.last_name, item.fathers_name);
                    employee_list.appendChild(option);
                });
            }
        };
        xhr.send(post_str({
            'sto_name': inp_sto_list.value
        }));
    }

    /**
     * Функція для отримання списку послуг
     *

```

```

    * @param void
    */
function get_task_list() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/get_task_list');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            let data = JSON.parse(xhr.response);
            task_list.innerHTML = '';
            data.forEach((item) => {
                let option =
document.createElement('option');
                option.value = item['service_name'];
                task_list.appendChild(option);
            });
        }
    };
    xhr.send();
}

/**
 * Функція для створення замовлення
 *
 * @param void
 */
function create_order() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/create_order');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

```

```

        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                order_id = data[0].id;
                document.getElementById('back').setAttribute('href', '/admin/order/delete/' + order_id)
            }
        };
        xhr.send(post_str({
            'client_id': client_id,
            'car_id': car_id,
        }));
    }

    /**
     * Функція для отримання списку завдань
     *
     * @param void
     */
    function get_task() {
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/admin/order/get_task/' + order_id);

        xhr.setRequestHeader('X-Requested-With', 'XMLHttpRequest');
        xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status != 200) {
                error.hidden = false;
                error.innerHTML = 'ERROR: ' + xhr.statusText;
            } else {
                error.hidden = true;
                let price = 0;
                let data = JSON.parse(xhr.response);
            }
        };
    }

```

```

        document.getElementById('table-
body').innerHTML = '';
        for (let i = 0; i < data.length; i++) {
            price += parseInt(data[i].price);
            const tr =
document.createElement('tr');
            tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>';
            tr.innerHTML += '<td>' +
data[i].service_name + '</td>';
            tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';
            tr.innerHTML += '<td>' +
data[i].price + '</td>';
            tr.innerHTML += '<td>' +
(data[i].is_done == 't' ? 'Tak' : 'Hi') + '</td>';
            tr.innerHTML += '<td>' + '<button
type="button" class="del btn" style="padding: 0;" id="' + i +
'"></button>' + '</td>';
            document.getElementById('table-
body').appendChild(tr);
        }

        let btns_del =
document.getElementsByClassName('del');
        for (let i = 0; i < btns_del.length;
i++) {
            btns_del[i].addEventListener('click'
, () => {
                if (confirm('Ви дійсно хочете
видалити цю послугу?')) {
                    let xhr = new
XMLHttpRequest();

```

```

        xhr.open('POST',
'/admin/order/delete_task/' + data[i].id);
        xhr.setRequestHeader('X-
Requested-With', 'XMLHttpRequest');
        xhr.setRequestHeader("Conten
t-type", "application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status != 200) {
                error.hidden =
false;

                error.innerHTML =
'ERROR: ' + xhr.statusText;

            } else {
                error.hidden = true;
                get_task();
            }
        };
        xhr.send();
    }
    });
}
};
xhr.send();
}
});

/**
 * Функція для склеювання об'єкту в URL строку
 *
 * @param {Object} data - об'єкт
 * @returns {String} - склеєна строка
 */
function post_str(data) {
    let str = '';
    for (let key in data) {

```



```

        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

```

```
/**
```

```
 * Функція для склеювання строк
```

```
 *
```

```
 * @param {String} split - розділювач
```

```
 * @param {...String} theArgs - строки
```

```
 * @returns {String} - склеєна строка
```

```
 */
```

```
function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

```

```
</script>
```

```
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
```

```
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
```

```
" crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

Файл `app/admin/order/view_edit.php`

```
<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
    <header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">
```

```

        <a href="/admin/order/index" class="d-flex align-items-
center mb-3 mb-md-0 me-md-auto link-body-emphasis text-
decoration-none" style="margin-left: 16px;">
            
            <span class="fs-4">Назад</span>
        </a>

        <button id="delete" type="button" class="btn btn-
outline-danger" style="margin: 0 8px;">Видалити</button>
        <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;">Зберегти</button>
    </header>
    <main>
        <div class="container">
            <div class="input-group mb-3">
                <span class="input-group-text">Замовник</span>
                <input type="text" class="form-control"
placeholder="Прізвище" value="<?= $order_info['first_name'] ?>
<?= $order_info['last_name'] ?> <?= $order_info['fathers_name']
?>" disabled>
            </div>

            <div class="input-group mb-3">
                <span class="input-group-text">Номер
телефону</span>
                <input type="text" class="form-control"
value="+<?= substr($order_info['phone_number'], 0, 3) ?> (<?=
substr($order_info['phone_number'], 3, 2) ?>) <?=
substr($order_info['phone_number'], 5, 3) ?>-<?=
substr($order_info['phone_number'], 8, 2) ?>-<?=
substr($order_info['phone_number'], 10, 2) ?>" disabled>
            </div>

            <div class="input-group mb-3">

```

```

        <span class="input-group-text">Дата
створення</span>
        <input type="date" class="form-control"
value="<?= $order_info['create_date'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">СТО</span>
        <input type="text" class="form-control"
value="<?= $sto_info['sto_name'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Автомобіль</span>
        <input type="text" class="form-control"
value="<?= $order_info['make'] ?> - <?= $order_info['model'] ?>
- <?= $order_info['license_plate'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Знижка
клієнта</span>
        <?php if ($order_info['discount'] == null) : ?>
            <input type="text" class="form-control"
value="0" disabled>
        <?php else : ?>
            <input type="text" class="form-control"
value="<?= $order_info['discount'] ?>" disabled>
        <?php endif; ?>
        <label class="input-group-text"
for="discount">%</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Знижка по
замовленню</span>

```

```

        <?php if ($order_info['order_discount'] == null)
: ?>

        <input id="discount" min="0" max="100"
type="number" class="form-control" value="0">
        <?php else : ?>
        <input id="discount" min="0" max="100"
type="number" class="form-control" value="<?=
$order_info['order_discount'] ?>">
        <?php endif; ?>
        <label class="input-group-text"
for="discount">%</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Статус</span>
        <select id="status" class="form-control"
value="" disabled>
            <? if ($order_info['completed'] == 't') : ?>
                <option value="0"
selected>Виконано</option>
                <option value="1">Не виконано</option>
            <? else : ?>
                <option value="0">Виконано</option>
                <option value="1" selected>Не
виконано</option>
            <? endif; ?>
        </select>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Оплата</span>
        <select class="form-control" disabled
autocomplete="off">
            <? if ($order_info['payment'] == 't') : ?>
                <option value="0"
selected>Оплачено</option>

```

```

        <option value="1">Не оплачено</option>
    <? else : ?>
        <option value="0">Оплачено</option>
        <option value="1" selected>Не
оплачено</option>

    <? endif; ?>
</select>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Вартість
послуг</span>
    <input id="price" type="text" class="form-
control" value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Сума
знижки</span>
    <input id="discount-price" type="text"
class="form-control" value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Вартість зі
знижкою</span>
    <input id="sum" type="text" class="form-control"
value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<p id="error" class="text-danger" hidden></p>
</div>

```

```

    <div class="container">
        <button id="new-task" type="button" class="btn btn-
primary" data-bs-toggle="modal" data-bs-target="#add-task-
model">Нове завдання</button>

        <div class="modal fade" id="add-task-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
                <div class="modal-content">
                    <div class="modal-header">
                        <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту клієнта</h1>
                        <button type="button" class="btn-
close" data-bs-dismiss="modal" aria-label="Close"></button>
                    </div>
                    <div class="modal-body">
                        <div class="input-group mb-3">
                            <span class="input-group-
text">Працівник</span>
                            <input id="employee" type="text"
class="form-control" list="employee-list">
                            <datalist id="employee-
list"></datalist>
                        </div>

                        <div class="input-group mb-3">
                            <span class="input-group-
text">Завдання</span>
                            <input id="task" type="text"
class="form-control" list="task-list">
                            <datalist id="task-
list"></datalist>
                        </div>
                    </div>
                </div>
            </div>
        </div>

```

```

        <div class="modal-footer">
            <output hidden id="error"
class="text-danger"></output>
            <button type="button" class="btn
btn-secondary" data-bs-dismiss="modal">Відміна</button>
            <button id="create" type="button"
class="btn btn-primary">Створити</button>
        </div>
    </div>
</div>
</div>
</div>
</div>

<div style="margin-left: 16px; margin-right: 16px;">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Назва послуги</th>
                <th scope="col">Виконавець</th>
                <th scope="col">Ціна</th>
                <th scope="col">Виконано</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody id="table-body"></tbody>
    </table>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const modal = new
bootstrap.Modal(document.getElementById('add-task-model'));
        const inp_discount =
document.getElementById('discount');

```



```

        const inp_status =
document.getElementById('status');
        const inp_price = document.getElementById('price');
        const inp_discount_price =
document.getElementById('discount-price');
        const inp_sum = document.getElementById('sum');
        const inp_employee =
document.getElementById('employee');
        const inp_task = document.getElementById('task');

        const employee_list =
document.getElementById('employee-list');
        const task_list = document.getElementById('task-
list');

        const btn_delete =
document.getElementById('delete');
        const btn_save = document.getElementById('save');
        const btn_new_task = document.getElementById('new-
task');
        const btn_create =
document.getElementById('create');

        const error = document.getElementById('error');
        const table_body = document.getElementById('table-
body');

        get_task();
        get_employee_list();
        get_task_list();

        /**
         * Функція для видалення замовлення
         *
         * @returns void
         */

```

```

        btn_delete.addEventListener('click', () => {
            if (confirm('Ви дійсно хочете видалити це
замовлення?')) {
                window.location.href =
'/admin/order/delete/' + '<?= $order_id ?>';
            }
        });

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_create.addEventListener('click', () => {
    let data = Object();
    data['name'] = inp_employee.value;
    data['task'] = inp_task.value;

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/add_task/' + '<?=
$order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;
        } else {
            error.hidden = true;
            modal.hide();
            get_task();
        }
    };
};

```

```

        xhr.send(post_str(data));
    });

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_save.addEventListener('click', () => {
    let is_error = false;
    let data = Object();
    data['discount'] = inp_discount.value;

    if (inp_discount.value == '') {
        data['discount'] = null;
    }

    if (inp_discount.value < 0 || inp_discount.value
> 100) {

        is_error = true;
        inp_discount.classList.add('is-invalid');
        return;
    }

    if (is_error) {
        error.hidden = false;
        error.innerHTML = 'Помилка вводу';
        return;
    }

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/update/' + '<?=$order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');

```

```

        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status !== 200) {
                error.hidden = false;
                error.innerHTML = 'ERROR: ' +
xhr.statusText;

                } else {
                    error.hidden = true;
                    window.location.href =
'/admin/order/index';
                }
            };

            xhr.send(post_str(data));
        });

/**
 * Функція для відображення модального вікна
 *
 * @returns void
 */
btn_new_task.addEventListener('click', () => {
    inp_employee.value = '';
    inp_task.value = '';
});

/**
 * Функція для відображення модального вікна
 *
 * @returns void
 */
inp_discount.addEventListener('change', () => {
    let price = parseInt(inp_price.value);
    let dis = (parseInt(inp_discount.value) / 100) +
(<?= $order_info['discount'] ?> / 100);

```

```

        if (dis > 1) {
            dis = 1;
        }
        dis = Math.round(price * dis);
        inp_discount_price.value = dis;
        inp_sum.value = price - dis;
    });

/**
 * Функція для отримання списку завдання
 *
 * @returns void
 */
function get_task() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/get_task/<?=$order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
        'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
        "application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status != 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
                xhr.statusText;
        } else {
            error.hidden = true;
            let price = 0;
            let data = JSON.parse(xhr.response);
            document.getElementById('table-
body').innerHTML = '';
            for (let i = 0; i < data.length; i++) {
                price += parseInt(data[i].price);
                const tr =
                    document.createElement('tr');

```

```

        tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>';

        tr.innerHTML += '<td>' +
data[i].service_name + '</td>';

        tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';

        tr.innerHTML += '<td>' +
data[i].price + '</td>';

        tr.innerHTML += '<td>' +
(data[i].is_done == 't' ? 'Tak' : 'Hi') + '</td>';

        tr.innerHTML += '<td>' + '<button
type="button" class="del btn" style="padding: 0;" id="' + i +
'"></button>' + '</td>';

        document.getElementById('table-
body').appendChild(tr);
    }

    inp_price.value = price;
    let dis = (parseInt(inp_discount.value)
/ 100) + (<?= $order_info['discount'] ?> / 100);
    if (dis > 1) {
        dis = 1;
    }
    dis = Math.round(price * dis);
    inp_discount_price.value = dis;
    inp_sum.value = price - dis;

    let btns_del =
document.getElementsByClassName('del');
    for (let i = 0; i < btns_del.length;
i++) {

        btns_del[i].addEventListener('click'
, () => {

```

```

        if (confirm('Ви дійсно хочете
видалити цю послугу?')) {

            let xhr = new
XMLHttpRequest();

            xhr.open('POST',
'/admin/order/delete_task/' + data[i].id);
            xhr.setRequestHeader('X-
Requested-With', 'XMLHttpRequest');
            xhr.setRequestHeader("Conten
t-type", "application/x-www-form-urlencoded");
            xhr.onload = () => {
                if (xhr.status != 200) {
                    error.hidden =
false;

                    error.innerHTML =
'ERROR: ' + xhr.statusText;

                } else {
                    error.hidden = true;
                    get_task();
                }
            };
            xhr.send();
        }
    });
}

};
xhr.send();
}

/**
 * Функція для отримання списку працівників
 *
 * @returns void
 */
function get_employee_list() {

```

```

        let xhr = new XMLHttpRequest();
        xhr.open('POST',
'/admin/order/get_employee_list');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                employee_list.innerHTML = '';
                data.forEach((item) => {
                    let option =
document.createElement('option');
                    option.value = sum_strs(' ',
item.first_name, item.last_name, item.fathers_name);
                    employee_list.appendChild(option);
                });
            }
        };
        xhr.send(post_str({
            'sto_name': '<?= $sto_info['sto_name'] ?>'
        }));
    }

/**
 * Функція для отримання списку завдань
 *
 * @returns void
 */
function get_task_list() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/get_task_list');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');

```



```

        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                task_list.innerHTML = '';
                data.forEach((item) => {
                    let option =
document.createElement('option');
                    option.value = item.service_name;
                    task_list.appendChild(option);
                });
            }
        };
        xhr.send();
    }
});

function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

/**
 * Функція для склеювання строк
 *
 * @param {String} split - розділювач
 * @param {...String} theArgs - строки
 * @returns {String} - склеєна строка
 */
function sum_strs(split, ...theArgs) {
    let total = '';

```

```

        for (const arg of theArgs) {
            if (arg == null) {
                continue;
            }
            total += arg + split;
        }
        return total.slice(0, -split.length);
    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/admin/order/view_index.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

```

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

<head>

<meta charset="utf-8">

<meta name="author" content="Dmytro Zamkovyi">

<title><?= \$title ?></title>

<link rel="shortcut icon" href="/data/favicon.ico"

type="image/png">

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEVDwWykc2MPK8M2HN

" crossorigin="anonymous">

</head>

<body>

<?= \$head ?>

<main style="margin-top: 16px;">

<div class="container as castrulla">

<button id="new-order" type="button" class="btn btn-primary">Нове замовлення</button>

<output <?= \$success ?> id="success" class="text-success">Замовлення успішно додано!</output>

</div>

<hr>

<div class="container accordion" id="according-search">

<div class="accordion-item">

<h2 class="accordion-header">

<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapse-search" aria-expanded="false" aria-controls="collapseTwo">

```

        <h1>Пошук</h1>
    </button>
</h2>
    <div id="collapse-search" class="accordion-
collapse collapse" data-bs-parent="#according-search">
        <div class="accordion-body">
            <div class="input-group mb-3">
                <span class="input-group-text">ПІВ
клієнта</span>

                <input id="search-first-name"
type="text" class="form-control" placeholder="Прізвище">
                <input id="search-last-name"
type="text" class="form-control" placeholder="Ім'я">
                <input id="search-father-name"
type="text" class="form-control" placeholder="Патронім">
            </div>

            <div class="input-group mb-3">
                <span class="input-group-text">Номер
телефону</span>

                <input id="search-phone-number"
type="text" class="form-control">
                <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
            </div>

            <div class="input-group mb-3">
                <span class="input-group-text">Номер
автомобіля</span>

                <input id="search-car" type="text"
class="form-control">
            </div>

            <div class="input-group mb-3">
                <span class="input-group-
text">Дата</span>

```

```

        <input id="search-start-date"
type="date" class="form-control">
        <label class="input-group-text">-
</label>
        <input id="search-end-date"
type="date" class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Статус</span>
        <select id="search-complete"
class="form-select" aria-label="Default select example">
            <option selected></option>
            <option
value="1">Завершено</option>
            <option value="2">В
процесі</option>
        </select>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Оплата</span>
        <select id="search-pay" class="form-
select" aria-label="Default select example">
            <option selected></option>
            <option
value="1">Оплачено</option>
            <option value="2">Не
оплачено</option>
        </select>
    </div>

    <div class="gap-2 d-md-flex justify-
content-md-end">

```

```

        <button id="search" type="button"
class="btn btn-primary">Пошук</button>
        <button id="search-reset"
type="button" class="btn btn-outline-primary">Скинути</button>
    </div>
</div>
</div>
</div>
</div>

<div style="margin-left: 16px; margin-right: 16px;">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">ПІВ клієнта</th>
                <th scope="col">Номер телефону</th>
                <th scope="col">Автомобіль</th>
                <th scope="col">Дата</th>
                <th scope="col">Статус</th>
                <th scope="col">Оплата</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody id="table-body"></tbody>
    </table>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const btn_create = document.getElementById('new-
order');
        const success = document.getElementById('success');

```

```

        const inp_search_first_name =
document.getElementById('search-first-name');
        const inp_search_last_name =
document.getElementById('search-last-name');
        const inp_search_father_name =
document.getElementById('search-father-name');
        const inp_search_phone_number =
document.getElementById('search-phone-number');
        const inp_search_car =
document.getElementById('search-car');
        const inp_search_start_date =
document.getElementById('search-start-date');
        const inp_search_end_date =
document.getElementById('search-end-date');
        const inp_search_complete =
document.getElementById('search-complete');
        const inp_search_pay =
document.getElementById('search-pay');
        const btn_search =
document.getElementById('search');
        const btn_search_reset =
document.getElementById('search-reset');

        const table_body = document.getElementById('table-
body');

        send();

        setTimeout(() => {
            success.hidden = true;
        }, 5000);

/**
 * Функція для перенаправлення на сторінку створення
замовлення
 *

```

```

        * @returns void
        */
btn_create.addEventListener('click', () => {
    window.location.href = '/admin/order/create';
});

/**
 * Функція для пошуку замовлень
 *
 * @returns void
 */
btn_search.addEventListener('click', () => {
    let search = {};
    let is_error = false;

    if (inp_search_first_name.value != '') {
        if (inp_search_first_name.value.length < 3
|| inp_search_first_name.value.length > 40) {
            inp_search_first_name.classList.add('is-
invalid');

            is_error = true;
        } else {
            inp_search_first_name.classList.remove('
is-invalid');

            search.first_name =
encodeURIComponent(inp_search_first_name.value);
        }
    } else {
        inp_search_first_name.classList.remove('is-
invalid');
    }

    if (inp_search_last_name.value != '') {
        if (inp_search_last_name.value.length < 3 ||
inp_search_last_name.value.length > 40) {

```



```

        inp_search_last_name.classList.add('is-
invalid');

        is_error = true;
    } else {
        inp_search_last_name.classList.remove('i
s-invalid');

        search.last_name =
encodeURIComponent(inp_search_last_name.value);
    }
} else {
    inp_search_last_name.classList.remove('is-
invalid');
}

    if (inp_search_father_name.value != '') {
        if (inp_search_father_name.value.length < 3
|| inp_search_father_name.value.length > 40) {
            inp_search_father_name.classList.add('is
-invalid');

            is_error = true;
        } else {
            inp_search_father_name.classList.remove(
'is-invalid');

            search.father_name =
encodeURIComponent(inp_search_father_name.value);
        }
    } else {
        inp_search_father_name.classList.remove('is-
invalid');
    }

    if (inp_search_phone_number.value != '') {
        let re = /^[ ]*\+?([0-9]{2})[-\([ ]*([0-
9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-
9]{2})[ ]*$/gm;

```

```

        let res =
re.exec(inp_search_phone_number.value);
        if (res) {
            search.phone_number =
encodeURIComponent(res.slice(1).join(''));
            inp_search_phone_number.classList.remove
('is-invalid');
        } else {
            inp_search_phone_number.classList.add('i
s-invalid');
            inp_search_phone_number.placeholder =
'Невірний формат номеру телефону'
            is_error = true;
        }
    } else {
        inp_search_phone_number.classList.remove('is
-invalid');
    }

    if (inp_search_car.value != '') {
        search.car =
encodeURIComponent(inp_search_car.value);
    }

    if (inp_search_start_date.value != '') {
        search.start_date =
encodeURIComponent(inp_search_start_date.value);
        inp_search_start_date.classList.remove('is-
invalid');
    }

    if (inp_search_end_date.value != '') {
        inp_search_end_date.classList.remove('is-
invalid');
        search.end_date =
encodeURIComponent(inp_search_end_date.value);

```

```

    }

    if (inp_search_start_date.value != '' &&
inp_search_end_date.value != '') {
        if (inp_search_start_date.value >
inp_search_end_date.value) {
            inp_search_start_date.classList.add('is-
invalid');

            inp_search_end_date.classList.add('is-
invalid');

            is_error = true;
        } else {
            inp_search_start_date.classList.remove('
is-invalid');

            inp_search_end_date.classList.remove('is
-invalid');
        }
    } else {
        inp_search_start_date.classList.remove('is-
invalid');

        inp_search_end_date.classList.remove('is-
invalid');
    }

    if (inp_search_complete.value != '') {
        switch (inp_search_complete.value) {
            case '1':
                search.complete =
encodeURIComponent('true');
                break;
            case '2':
                search.complete =
encodeURIComponent('false');
                break;
        }
    }
}

```

```

        if (inp_search_pay.value != '') {
            switch (inp_search_pay.value) {
                case '1':
                    search.pay =
encodeURIComponent('true');
                    break;
                case '2':
                    search.pay =
encodeURIComponent('false');
                    break;
            }
        }

        send(search);
    });

/**
 * Функція для скидання параметрів пошуку
 *
 * @returns void
 */
btn_search_reset.addEventListener('click', () => {
    inp_search_first_name.value = '';
    inp_search_last_name.value = '';
    inp_search_father_name.value = '';
    inp_search_phone_number.value = '';
    inp_search_car.value = '';
    inp_search_start_date.value = '';
    inp_search_end_date.value = '';
    inp_search_complete.value = '';
    inp_search_pay.value = '';
});

/**
 * Функція для відправки запиту на сервер

```

```

*
* @param {Object} search - об'єкт з параметрами
пошуку
* @returns void
*/
function send(search = {}) {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/order/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
    xhr.onload = () => {
        if (xhr.status == 200) {
            table_body.innerHTML = '';
            let data = JSON.parse(xhr.responseText);
            for (let i = 0; i < data.length; i++) {
                let tr =
document.createElement('tr');
                tr.classList.add('col-12', 'col-lg-
auto', 'my-2', 'justify-content-center', 'my-md-0');
                tr.innerHTML = '<th score="row">' +
(i + 1) + '</th>';
                tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';
                tr.innerHTML += '<td>' + '+' +
data[i].phone_number.slice(0, 3) + ' (' +
data[i].phone_number.slice(3, 5) + ') ' +
data[i].phone_number.slice(5, 8) + '-' +
data[i].phone_number.slice(8, 10) + '-' +
data[i].phone_number.slice(10, 12) + '</td>';
                tr.innerHTML += '<td>' + sum_strs('
- ', data[i].make, data[i].model, data[i].license_plate) +
'</td>';

```

```

        tr.innerHTML += '<td>' +
data[i].create_date + '</td>';
        tr.innerHTML += '<td>' +
(data[i].completed ? 'Завершено' : 'В процесі') + '</td>';
        tr.innerHTML += '<td>' +
(data[i].payment ? 'Оплачено' : 'Не оплачено') + '</td>';
        tr.innerHTML += '<td>' + '<button
type="button" class="edit btn" style="padding: 0;"
onclick=edit(' + data[i]['id'] + ')></button>'
+ '</td>'

        table_body.appendChild(tr);
    }
}
};
xhr.send(post_str(search));
}
});

/**
 * Функція для перетворення об'єкта в строку
 *
 * @param {Object} obj - об'єкт
 * @returns {String} - строка
 */
function post_str(obj) {
    let str = '';
    for (let p in obj) {
        str += encodeURIComponent(p) + "=" +
encodeURIComponent(obj[p]);
        str += '&';
    }
    return str.slice(0, -1);
}

```

```

/**
 * Функція для склеювання строк
 *
 * @param {String} split - розділювач
 * @param {...String} theArgs - строки
 * @returns {String} - склеєна строка
 */
function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

/**
 * Функція для перенаправлення на сторінку редагування
працівника
 *
 * @param {Number} i - id працівника
 * @returns void
 */
function edit(i) {
    window.location.href = '/admin/order/edit/' + i;
}
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

```

</html>

Файл app/admin/service/action.php

<?

```
/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */
```

```
class adminService extends Curswork\ActionClass
{
    /**
     * Контроллер для виводу сторінки зі списком послуг
     *
     * @param $params - параметри запиту
     * @param $data - дані запиту
     * @return mixed
     */
    public function action_index($params, $data)
    {
        $view = "index";
        $data['label'] = 'Перегляд послуг';
        $data['menu'] = array(
```



```

        array(
            'name' => 'Працівники',
            'href' => '/admin/employee/index',
            'active' => false,
            'img' => '/data/icon/employee.png'
        ),
        array(
            'name' => 'Послуги',
            'href' => '/admin/service/index',
            'active' => true,
            'img' => '/data/icon/service_activ.png'
        ),
        array(
            'name' => 'Замовлення',
            'href' => '/admin/order/index',
            'active' => false,
            'img' => '/data/icon/order.png'
        ),
        array(
            'name' => 'Клієнти',
            'href' => '/admin/client/index',
            'active' => false,
            'img' => '/data/icon/client.png'
        ),
        array(
            'name' => 'Вихід',
            'href' => '/index/index/logout',
            'active' => false,
            'img' => '/data/icon/logout.png'
        ),
    );
    $data['head'] = $this->load('index', 'index', 'head',
'', $data);
    $data['title'] = 'Послуги · ' . $this->config-
>projectName;

```

```

        $this->view->out($view, $data);
    }

    /**
     * Контроллер для створення нової послуги
     *
     * @param $params - параметри запиту
     * @param $data - дані запиту
     * @return mixed
     */
    public function action_create($params, $data)
    {
        $data['name'] = urldecode($this->request->post->name);
        $data['description'] = urldecode($this->request->post->description);
        $data['price'] = (float)urldecode($this->request->post->price);

        if ($this->model->create($data)) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }

    /**
     * Контроллер для пошуку послуги
     *
     * @param $params - параметри запиту
     * @param $data - дані запиту
     * @return mixed
     */
    public function action_search($params, $data)
    {
        $data['search'] = urldecode($this->request->post->search);
    }

```

```

        $result = $this->model->search($data);

        if ($result !== false) {
            header('HTTP/1.1 200 OK');
            echo json_encode($result);
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
    }

/**
 * Контроллер для видалення послуги
 *
 * @param $params - параметри запиту
 * @param $data - дані запиту
 * @return mixed
 */
public function action_delete($params, $data)
{
    $data['service_id'] = (int)$params;

    $this->model->delete($data);
    header('HTTP/1.1 200 OK');
    header('Location: /admin/service/index');
}

/**
 * Контроллер для оновлення послуги
 *
 * @param $params - параметри запиту
 * @param $data - дані запиту
 * @return mixed
 */
public function action_update($params, $data)
{

```

```

        $data['service_id'] = urldecode($this->request->post->id);
        $data['name'] = urldecode($this->request->post->name);
        $data['description'] = urldecode($this->request->post->description);
        $data['price'] = (float)urldecode($this->request->post->price);

        $this->model->update($data);
        header('HTTP/1.1 200 OK');
    }

    /**
     * Контроллер для виводу сторінки редагування послуги
     *
     * @param $params - параметри запиту
     * @param $data - дані запиту
     * @return mixed
     */
    public function action_edit($params, $data)
    {
        $data['service_id'] = (int)$params;
        $view = "edit";
        $data['title'] = 'Товар · ' . $this->config->projectName;

        $data['service'] = $this->model->get_service($data['service_id']);

        $this->view->out($view, $data);
    }
}

```

Файл app/admin/service/model.php

<?

```
/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */
```

```
class adminServiceModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для додавання нової послуги
     *
     * @param $data - дані для додавання
     * @return bool
     */
    public function create($data)
    {
        $sql = 'INSERT INTO "service" (service_name,
description, price) VALUES (\'' . $this->db-
>escape($data['name']) . '\', \'' . $this->db-
>escape($data['description']) . '\', ' . $this->db-
>escape($data['price']) . ');';
        var_dump($sql);
        $this->db->query($sql);
    }
}
```

```

        if ($this->db->last_error) {
            return false;
        } else {
            return true;
        }
    }
}

/**
 * Метод моделі для пошуку послуг
 *
 * @param $data - дані для пошуку
 * @return mixed
 */
public function search($data)
{
    $sql = 'SELECT * FROM "service" WHERE service_name LIKE
\'%\' . $this->db->escape($data['search']) . '%\' OR description
LIKE \'%\' . $this->db->escape($data['search']) . '%\';';
    $result = $this->db->query($sql);

    if ($this->db->last_error) {
        return false;
    } else {
        return $result;
    }
}

/**
 * Метод моделі для отримання всіх послуг
 *
 * @param $id - id послуги
 * @return mixed
 */
public function get_service($id)
{

```

```

        $sql = 'SELECT * FROM "service" WHERE id = ' . $this->db->escape($id) . ' ';
        $result = $this->db->query($sql);

        if ($this->db->last_error) {
            return false;
        } else {
            return $result[0];
        }
    }

    /**
     * Метод моделі для видалення послуги
     *
     * @param $data - дані для видалення
     * @return mixed
     */
    public function delete($data)
    {
        $sql = 'DELETE FROM "service" WHERE id = ' . $this->db->escape($data['service_id']) . ' ';
        $this->db->query($sql);
    }

    /**
     * Метод моделі для оновлення послуги
     *
     * @param $data - дані для оновлення
     * @return mixed
     */
    public function update($data)
    {
        $sql = 'UPDATE "service" SET service_name = \'' . $this->db->escape($data['name']) . '\', description = \'' . $this->db->escape($data['description']) . '\', price = ' . $this->db-

```

```

>escape($data['price']) . ' WHERE id = ' . $this->db-
>escape($data['service_id']) . ' ';
    $this->db->query($sql);
    }
}

```

Файл app/admin/service/view_edit.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-

```



```

T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
  <header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">

    <a href="/admin/service/index" class="d-flex align-
items-center mb-3 mb-md-0 me-md-auto link-body-emphasis text-
decoration-none" style="margin-left: 16px;">
      
      <span class="fs-4">Назад</span>
    </a>

    <h4 class="d-flex align-items-center mb-3 mb-md-0 me-md-
auto link-body-emphasis text-decoration-none">Редагування
послуги</h4>

    <button id="delete" type="button" class="btn btn-
outline-danger" style="margin: 0 8px;">Видалити</button>
    <button id="escape" type="button" class="btn btn-
outline-secondary" style="margin: 0 8px;">Скасувати</button>
    <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;">Зберегти</button>
  </header>
  <main class="container">
    <div>

      <div class="input-group mb-3">
        <span class="input-group-text">Назва</span>
        <input id="name" type="text" class="form-
control" value="<?= $service['service_name'] ?>">
      </div>

```

```

    <div class="input-group mb-3">
      <span class="input-group-text">Опис</span>
      <textarea id="description" class="form-
control"><?= $service['description'] ?></textarea>
    </div>

    <div class="input-group mb-3">
      <span class="input-group-text">Ціна</span>
      <input id="price" type="text" class="form-
control" value="<?= $service['price'] ?>">
      <label class="input-group-text">€</label>
      <label class="input-group-text">0.00</label>
    </div>

    <p id="error" class="text-danger" hidden></p>
    <p id="success" class="text-success" hidden>Данні
успішно збережені</p>
  </div>
</main>

<script>
  window.addEventListener('DOMContentLoaded', () => {
    const inp_name = document.getElementById('name');
    const inp_description =
document.getElementById('description');
    const inp_price = document.getElementById('price');

    const btn_delete =
document.getElementById('delete');
    const btn_cancel =
document.getElementById('escape');
    const btn_save = document.getElementById('save');

    const error = document.getElementById('error');
    const success = document.getElementById('success');

```

```

/**
 * Функція для повернення на попередню сторінку
 *
 * @returns void
 */
btn_cancel.addEventListener('click', () => {
    window.location.href = '/admin/service/index';
});

/**
 * Функція для видалення послуги
 *
 * @returns void
 */
btn_delete.addEventListener('click', () => {
    if (confirm('Ви дійсно хочете видалити цього
працівника?')) {
        window.location.href =
'/admin/service/delete/' + '<?= $service['id'] ?>';
    }
});

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_save.addEventListener('click', () => {
    let is_error = false;
    let data = {
        id: '<?= $service['id'] ?>',
        name: inp_name.value,
        description: inp_description.value,
        price: inp_price.value
    };
};

```

```

    if (data.name.length < 3 || data.name.length >
40) {

        is_error = true;
        inp_name.classList.add('is-invalid');
    } else {
        inp_name.classList.remove('is-invalid')
    }

    if (data.price <= 0) {
        is_error = true;
        inp_price.classList.add('is-invalid');
    } else {
        data.price = parseFloat(data.price);
        if (isNaN(data.price) || data.price <= 0) {
            is_error = true;
            inp_price.classList.add('is-invalid');
        } else {
            inp_price.classList.remove('is-
invalid');

        }
    }

    if (is_error) {
        error.hidden = false;
        return;
    } else {
        error.hidden = true;
    }

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/service/update/<?=$service['id'] ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

```

```

        xhr.onload = () => {
            if (xhr.status !== 200) {
                document.getElementById('error').hidden
= false;

                document.getElementById('error').innerHT
ML = 'ERROR: ' + xhr.statusText;
            } else {
                document.getElementById('error').hidden
= true;

                document.getElementById('success').hidde
n = false;

                setTimeout(() => {
                    window.location.href =
'/admin/service/index';

                    }, 5000);
                }
            };

            xhr.send(post_str(data));
        });
    })

/**
 * Функція для перетворення об'єкту в строку
 *
 * @param {object} data - об'єкт для перетворення
 * @returns {string} - строка з даними
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

```

```

    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/admin/service/view_index.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">
<!--
License

```

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

```
-->
```

```

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">

```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
  <?= $head ?>
  <main style="margin-top: 16px;">
    <div class="container">
      <div style="margin-bottom: 16px;">
        <button id="new_service" type="button"
class="btn btn-primary" data-bs-toggle="modal" data-bs-
target="#add-service">Нова послуга</button>
        <output hidden id="success" class="text-
success">Послугу успішно додано!</output>

        <div class="modal fade" id="add-service" data-
bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"
aria-labelledby="staticBackdropLabel" aria-hidden="true">
          <div class="modal-dialog modal-dialog-
centered modal-lg modal-dialog-scrollable">
            <div class="modal-content">
              <div class="modal-header">
                <h1 class="modal-title fs-5"
id="staticBackdropLabel">Додавання послуги</h1>
                <button type="button"
class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
              </div>
              <div class="modal-body">
                <div class="input-group mb-3">
                  <span class="input-group-
text">Назва</span>

```

```

        <input id="name" type="text"
class="form-control">

    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Опис</span>

        <textarea id="description"
class="form-control"></textarea>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Ціна</span>

        <input id="price"
type="text" class="form-control">

        <label class="input-group-
text">€</label>

        <label class="input-group-
text">0.00</label>
    </div>
</div>
<div class="modal-footer">
    <output hidden id="error"
class="text-danger"></output>

    <button type="button" class="btn
btn-secondary" data-bs-dismiss="modal">Відміна</button>
    <button id="create"
type="button" class="btn btn-primary">Створити</button>
</div>
</div>
</div>
<div class="input-group mb-3">

```



```

        <input id="input-search" type="text"
class="form-control" aria-describedby="search">
        <button class="btn btn-primary" type="button"
id="search">Пошук</button>
    </div>
    <div style="margin-top: 16px;">
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">#</th>
                    <th scope="col">Назва</th>
                    <th scope="col">Ціна</th>
                    <th scope="col">Опис</th>
                </tr>
            </thead>
            <tbody id="table-body" class="table-group-
divider"></tbody>
        </table>
    </div>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const modal = new
bootstrap.Modal(document.getElementById('add-service'));
        const success = document.getElementById('success');
        const error = document.getElementById('error');

        const inp_name = document.getElementById('name');
        const inp_description =
document.getElementById('description');
        const inp_price = document.getElementById('price');

        const btn_create =
document.getElementById('create');

```

```

        const btn_new_service =
document.getElementById('new_service');

        const inp_search = document.getElementById('input-
search');

        const btn_search =
document.getElementById('search');

        const table_body = document.getElementById('table-
body');

        search();

        /**
         * Функція для очищення полів вводу при додаванні
нової послуги
         *
         * @returns void
         */
        btn_new_service.addEventListener('click', () => {
            inp_name.value = '';
            inp_description.value = '';
            inp_price.value = '';
        });

        /**
         * Функція для додавання нової послуги
         *
         * @returns void
         */
        btn_create.addEventListener('click', () => {
            let is_error = false;

            let data = {
                name: inp_name.value,
                description: inp_description.value,

```

```

        price: inp_price.value
    };

    if (data.name.length < 3 || data.name.length >
40) {

        inp_name.classList.add('is-invalid');
        inp_name.placeholder = 'Введіть назву
товару';

        is_error = true;
    } else {
        inp_name.classList.remove('is-invalid');
    }

    if (data.price <= 0) {
        inp_price.classList.add('is-invalid');
        inp_price.placeholder = 'Введіть ціну
товару';

        is_error = true;
    } else {
        data.price = parseFloat(data.price);
        if (isNaN(data.price)) {
            inp_price.classList.add('is-invalid');
            inp_price.placeholder = 'Введіть ціну
товару';

            is_error = true;
        } else {
            if (data.price <= 0) {
                inp_price.classList.add('is-
invalid');

                inp_price.placeholder = 'Введіть
ціну товару';

                is_error = true;
            } else {
                inp_price.classList.remove('is-
invalid');
            }
        }
    }
}

```

```

    }
}

if (is_error) {
    error.hidden = false;
    error.innerHTML = 'Заповніть всі поля!';
    return;
} else {
    error.hidden = true;
}

let xhr = new XMLHttpRequest();
xhr.open('POST', '/admin/service/create');
xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
xhr.onload = () => {
    if (xhr.status == 200) {
        success.hidden = false;
        setTimeout(() => {
            success.hidden = true;
        }, 5000);
        modal.hide();
        search();
    } else {
        error.hidden = false;
        error.innerHTML = xhr.response;
    }
};
xhr.send(post_str(data));
});

/**
 * Функція для пошуку послуги
 *

```

```

        * @returns void
        */
        btn_search.addEventListener('click', search);

/**
 * Функція для надсилання запиту на пошук послуги
 *
 * @returns void
 */
function search() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/admin/service/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            let result =
JSON.parse(xhr.responseText);
            table_body.innerHTML = '';
            for (let i = 0; i < result.length; i++)
{
                let tr =
document.createElement('tr');
                tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>'
                tr.innerHTML += '<td>' +
result[i].service_name + '</td>'
                tr.innerHTML += '<td>' +
result[i].price + '</td>'
                tr.innerHTML += '<td>' +
result[i].description + '</td>'
                tr.innerHTML += '<td>' + '<button
type="button" class="edit btn" style="padding: 0;"
onclick=edit(' + result[i]['id'] + ')></button>'
+ '</td>'

        table_body.appendChild(tr);
    }
}

};

xhr.send(post_str({
    "search": inp_search.value
}));

}

});

/**
 * Функція для перетворення об'єкту в строку для
відправки URL-запитом
 *
 * @param data - об'єкт для перетворення
 * @returns string
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

/**
 * Функція для переходу на сторінку редагування послуги
 *
 * @param id - id послуги
 * @returns void
 */
function edit(id) {
    window.location.href = '/admin/service/edit/' + id;

```

```

    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл `app/employee/detail/action.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class employeeDetail extends Curswork\ActionClass
{
    /**
     * Контроллер головної сторінки працівника вкладки "Деталі"
     *

```

```

* @param $params - параметри запиту
* @param $data - дані, що передаються в шаблон
* @return void
*/
public function action_index($params, $data)
{
    $view = "index";
    $data['label'] = 'Доступні деталі';
    $data['menu'] = array(
        array(
            'name' => 'Завдання',
            'href' => '/employee/task/index',
            'active' => false,
            'img' => '/data/icon/task.png'
        ),
        array(
            'name' => 'Деталі',
            'href' => '/employee/detail/index',
            'active' => true,
            'img' => '/data/icon/detail_activ.png'
        ),
        array(
            'name' => 'Розклад',
            'href' => '/employee/index/schedule',
            'active' => false,
            'img' => '/data/icon/calendar.png'
        ),
        array(
            'name' => 'Вихід',
            'href' => '/index/index/logout',
            'active' => false,
            'img' => '/data/icon/logout.png'
        ),
    );
    $data['head'] = $this->load('index', 'index', 'head',
'', $data);

```



```

        $data['title'] = 'Деталі · ' . $this->config-
>projectName;

        $this->view->out($view, $data);
    }

    /**
     * Контроллер пошуку деталей
     *
     * @param $params - параметри запиту
     * @param $data - дані, що передаються в шаблон
     * @return void
     */
    public function action_search($params, $data)
    {
        $data['search'] = urldecode($this->request->post-
>search);

        echo json_encode($this->model-
>get_details($data['search']));
    }

    /**
     * Контроллер отримання деталі
     *
     * @param $params - параметри запиту
     * @param $data - дані, що передаються в шаблон
     * @return void
     */
    public function action_get($params, $data)
    {
        $data['search'] = urldecode($this->request->post-
>search);

        echo json_encode($this->model-
>get_detail($data['search']));
    }

```

```

/**
 * Контроллер відправки замовлення
 *! В контролері відбувається симуляція відправки
замовлення на сторонній сервіс
 *
 * @param $params - параметри запиту
 * @param $data - дані, що передаються в шаблон
 * @return void
 */
public function action_order($params, $data)
{
    $data['data'] = urldecode($this->request->post->data);
    // Симуляці відправлення замовлення на сторонній сервіс
    echo $data['data'];
}
}

```

Файл `app/employee/detail/model.php`

<?

```

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

```

```

class employeeDetailModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для пошуку серед деталей
     *
     * @param string $search - пошуковий запит
     * @return array
     */
    public function get_details($search)
    {
        $sql = 'SELECT * FROM "detail" WHERE detail_name LIKE
\'%\' . $this->db->escape($search) . \'%\'' LIMIT 6;';
        return $this->db->query($sql);
    }

    /**
     * Метод моделі для отримання даних деталі
     *
     * @param string $search - пошуковий запит
     * @return array
     */
    public function get_detail($search)
    {
        $sql = 'SELECT * FROM "detail" WHERE detail_name = \'' .
$this->db->escape($search) . '\';';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        } else {
            return $result[0];
        }
    }
}

```

Файл `app/employee/detail/view_index.php`

```
<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
    <?= $head ?>
    <main style="margin-top: 8px;">
```

```

<div class="container">
  <div class="input-group mb-3">
    <input id="search" type="text" class="form-
control" list="search-list" placeholder="Пошук">
    <datalist id="search-list"></datalist>
    <button class="btn btn-primary" type="button"
id="add-to-order">Додати</button>
  </div>

  <div>
    <table class="table table-striped">
      <thead>
        <tr>
          <th scope="col">#</th>
          <th scope="col">Назва</th>
          <th scope="col">Кількість</th>
          <th scope="col">Ціна, за штуку</th>
          <th scope="col">Ціна, всього</th>
        </tr>
      </thead>
      <tbody id="table-body"></tbody>
    </table>
    <button id="order" class="btn btn-
success">Замовити</button>
    <output id="total-price">Всього: 0</output>
  </div>
</div>
</main>

<script>
  let data = Object();
  window.addEventListener('DOMContentLoaded', () => {
    const inp_search =
document.getElementById('search');
    const search_list = document.getElementById('search-
list');

```

```

        const btn_add = document.getElementById('add-to-
order');

        const table_body = document.getElementById('table-
body');

        const out_total_price =
document.getElementById('total-price');

        const btn_order = document.getElementById('order');

/**
 * Функція відправки запиту на сервер для пошукових
запитів
 *
 * @return void
 */
inp_search.addEventListener('input', () => {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/employee/detail/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            let data = JSON.parse(xhr.response);
            search_list.innerHTML = '';
            data.forEach((item) => {
                let option =
document.createElement('option');
                option.value = item['detail_name'];
                search_list.appendChild(option);
            });
        }
    };
    xhr.send('search=' +
encodeURIComponent(inp_search.value));
});

```

```

    /**
     * Функція відправки запиту на сервер для отримання
даних деталі
     *
     * @return void
     */
    btn_add.addEventListener('click', () => {
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/employee/detail/get');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let res = JSON.parse(xhr.responseText);
                if (data[res['id']]) {
                    data[res['id']]['count']++;
                } else {
                    data[res['id']] = {
                        'name': res['detail_name'],
                        'count': 1,
                        'price':
parseFloat(res['price'].slice(1).replace(',', ''))
                    };
                }
            }
            render(data);
        };
        xhr.send('search=' +
encodeURIComponent(inp_search.value));
    });

    /**
     * Функція рендерингу таблиці

```

```

*
* @param {Array} rdata - дані для рендерингу
* @return void
*/
function render(rdata) {
    table_body.innerHTML = '';
    let count = 0;
    let total_price = 0;
    for (let i in rdata) {
        if (rdata[i]['count'] == 0) {
            delete rdata[i];
            continue;
        }
        total_price += rdata[i]['price'] *
rdata[i]['count'];
        count++;
        let tr = document.createElement('tr');
        tr.classList.add('col-12', 'col-lg-auto',
'my-2', 'justify-content-center', 'my-md-0');
        tr.innerHTML += '<th scope="row">' + count +
'</th>';
        tr.innerHTML += '<td>' + rdata[i]['name'] +
'</td>';
        tr.innerHTML += '<td>' + '<input id="count-'
+ i +'" class="inp-count" type="number" min=0 value="' +
rdata[i]['count'] + '></td>';
        tr.innerHTML += '<td>' + rdata[i]['price'] +
'</td>';
        tr.innerHTML += '<td>' + rdata[i]['price'] *
rdata[i]['count'] + '</td>';
        table_body.appendChild(tr);
    }
    out_total_price.value = 'Всього: ' +
total_price;

```



```

        let inp_count =
document.getElementsByClassName('inp-count');
        for (let i = 0; i < inp_count.length; i++) {
            let key = inp_count[i].id.split('-')[1];
            inp_count[i].addEventListener('input', () =>
{
                rdata[key]['count'] =
inp_count[i].value;
                render(data);
            });
        }
        data = rdata;
    }

/**
 * Функція відправки запиту на сервер для оформлення
замовлення
 *
 * @return void
 */
    btn_order.addEventListener('click', () => {
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/employee/detail/order');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let res = JSON.parse(xhr.responseText);
                alert('Замовлення успішно оформлено');
                render(Object());
            } else {
                alert('Помилка оформлення замовлення');
            }
        };
    });

```

```

        let send_data = Object();
        let count = 0;
        for (let i in data) {
            send_data[count] = Object();
            send_data[count]['id'] = i;
            send_data[count]['count'] =
data[i]['count'];
            count++;
        }
        xhr.send('data=' +
encodeURIComponent(JSON.stringify(send_data)));
    });
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/employee/index/action.php

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.

```

```

* - Copying or reproducing any part of this code in any form.
* - Modifying or altering any part of this code.
*
* See the full text of the license in the LICENSE file in the
root directory of this repository.
*/

```

```

class employeeIndex extends Curswork\ActionClass
{
    /**
     * Контролер для перенаправлення на головну сторінку
працівника
     *
     * @return void
     */
    public function action_index()
    {
        $this->response->redirect('/employee/task/index');
    }

    /**
     * Контролер для відображення розкладу сторінки працівника
     *
     * @param array $params - параметри
     * @param array $data - дані
     * @return void
     */
    public function action_schedule($params, $data)
    {
        $view = "schedule";

        $data['label'] = 'Мій графік';
        $data['menu'] = array(
            array(
                'name' => 'Завдання',
                'href' => '/employee/task/index',

```

```

        'active' => false,
        'img' => '/data/icon/task.png'
    ),
    array(
        'name' => 'Деталі',
        'href' => '/employee/detail/index',
        'active' => false,
        'img' => '/data/icon/detail.png'
    ),
    array(
        'name' => 'Розклад',
        'href' => '/employee/index/schedule',
        'active' => true,
        'img' => '/data/icon/calendar_activ.png'
    ),
    array(
        'name' => 'Вихід',
        'href' => '/index/index/logout',
        'active' => false,
        'img' => '/data/icon/logout.png'
    ),
);
$data['head'] = $this->load('index', 'index', 'head',
'', $data);

    $data['title'] = 'Розклад · ' . $this->config-
>projectName;
    $data['schedule'] = $this->load('index', 'schedule',
'index');

    $this->view->out($view, $data);
}
}

```

Файл app/employee/index/view_schedule.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
  <meta charset="utf-8">
  <meta name="author" content="Dmytro Zamkovyi">
  <title><?= $title ?></title>
  <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
  <?= $head ?>
  <?= $schedule ?>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst

```

```

rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл `app/employee/task/action.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class employeeTask extends Curswork\ActionClass
{
    /**
     * Контроллер головної сторінки працівника вкладки
"Завдання"
     *
     * @param $params - параметри запиту
     * @param $data - дані, що передаються в шаблон
     * @return void
     */
}

```

```

public function action_index($params, $data)
{
    $view = "index";
    $data['label'] = 'Мої завдання';
    $data['menu'] = array(
        array(
            'name' => 'Завдання',
            'href' => '/employee/task/index',
            'active' => true,
            'img' => '/data/icon/task_activ.png'
        ),
        array(
            'name' => 'Деталі',
            'href' => '/employee/detail/index',
            'active' => false,
            'img' => '/data/icon/detail.png'
        ),
        array(
            'name' => 'Розклад',
            'href' => '/employee/index/schedule',
            'active' => false,
            'img' => '/data/icon/calendar.png'
        ),
        array(
            'name' => 'Вихід',
            'href' => '/index/index/logout',
            'active' => false,
            'img' => '/data/icon/logout.png'
        ),
    );
    $data['head'] = $this->load('index', 'index', 'head',
'', $data);
    $data['title'] = 'Завдання · ' . $this->config-
>projectName;

    $this->view->out($view, $data);

```

```

    }

    /**
     * Контроллер пошуку завдань
     *
     * @param $params - параметри запиту
     * @param $data - дані, що передаються в шаблон
     * @return void
     */
    public function action_search($params, $data)
    {
        $data['search'] = urldecode($this->request->post->search);
        echo json_encode($this->model->search($data['search']));
    }

    /**
     * Контроллер відмітки завдання як виконаного
     *
     * @param $params - параметри запиту
     * @param $data - дані, що передаються в шаблон
     * @return void
     */
    public function action_complete($params, $data)
    {
        $data['id'] = (int)$params;
        if ($this->model->complete($data['id'])) {
            header('HTTP/1.1 200 OK');
        } else {
            header('HTTP/1.1 400 Bad Request');
        }
        echo $data['id'];
    }
}

```


Файл app/employee/task/model.php

```
<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class employeeTaskModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі отримання завдань працівника
     *
     * @param string $search - рядок для пошуку
     * @return array|bool
     */
    public function search($search)
    {
        $sql = 'SELECT task.id, car.make, car.model,
car.license_plate, service.service_name FROM task, "order",
service, car WHERE task.id_order = "order".id AND
task.id_service = service.id AND "order".id_car = car.id AND
(service.service_name LIKE \'%\' . $this->db->escape($search) .
\'%\'' OR service.description LIKE \'%\' . $this->db-
>escape($search) . \'%\'' OR car.license_plate LIKE \'%\' . $this-
```

```

>db->escape($search) . '%\' OR car.model LIKE \'%' . $this->db-
>escape($search) . '%\' OR car.make LIKE \'%' . $this->db-
>escape($search) . '%\' AND task.is_done = false AND
task.id_employee = ' . $this->session->user_id . ' ';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    } else {
        return $result;
    }
}

/**
 * Метод моделі відмітки завдання як виконаного
 *
 * @param int $id - ідентифікатор завдання
 * @return bool
 */
public function complete($id)
{
    $sql = 'UPDATE task SET is_done = true WHERE id = ' .
$this->db->escape($id) . ' AND id_employee = ' . $this->session-
>user_id . ' ';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    } else {
        $this->check_order($id);
        return true;
    }
}

/**
 * Метод моделі перевірки виконання всіх завдань замовлення
 *
 * @param int $task_id - ідентифікатор завдання

```

```

        * @return void
        */
        public function check_order($task_id)
        {
            $sql = 'SELECT id_order FROM task WHERE id = ' . $this->db->escape($task_id) . ' ';
            $order_id = $this->db->query($sql);

            if (!$this->db->last_error) {
                $sql = 'SELECT id, is_done FROM task WHERE id_order = ' . $this->db->escape($order_id[0]['id_order']) . ' ';
                $result = (array)$this->db->query($sql);
                if (!$this->db->last_error) {
                    $is_done = true;
                    foreach ($result as $task) {
                        if ($task['is_done'] == 'f') {
                            $is_done = false;
                        }
                    }
                    if ($is_done) {
                        $sql = 'UPDATE "order" SET completed = true WHERE id = ' . $this->db->escape($order_id[0]['id_order']) . ' ';
                        $result = $this->db->query($sql);
                    }
                }
            }
        }
    }
}

```

Файл `app/employee/task/view_index.php`

```

<!DOCTYPE html>
<html data-bs-theme="dark">

```

<!--

License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

<head>

<meta charset="utf-8">

<meta name="author" content="Dmytro Zamkovyi">

<title><?= \$title ?></title>

<link rel="shortcut icon" href="/data/favicon.ico"

type="image/png">

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

T3c6CoIi6uLrA9TneNEoa7RxnatzjcdSCmG1MXxSR1GAsXEVDwvykc2MPK8M2HN

" crossorigin="anonymous">

</head>

<body>

<?= \$head ?>

<main style="margin-top: 16px;">

<div class="container">

<div class="input-group mb-3">

<input id="input-search" type="text"

class="form-control" aria-describedby="search">

```

        <button class="btn btn-primary" type="button"
id="search">Пошук</button>
    </div>
    <table class="table">
        <thead>
            <tr>
                <th>#</th>
                <th>Автомобіль</th>
                <th>Завдання</th>
                <th style="width: 60px;">Виконано</th>
            </tr>
        </thead>
        <tbody id="table"></tbody>
    </table>
</div>
</main>

<script>
    let data = Array();
    window.addEventListener('DOMContentLoaded', () => {
        const inp_search = document.getElementById('input-
search');
        const btn_search =
document.getElementById('search');
        const table = document.getElementById('table');

        const check =
document.getElementsByClassName('checkbox');

        send();
        setInterval(send, 60000);

        btn_search.addEventListener('click', send);

/**
 * Функція відправки запиту на сервер

```

```

*
* @return void
*/
function send() {
    const search = inp_search.value;
    const xhr = new XMLHttpRequest();
    xhr.open('POST', '/employee/task/search');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = function() {
        if (xhr.status == 200) {
            let result = JSON.parse(xhr.response);
            if (result !== data) {
                data = result;
                render(data);
            }
        }
    };
    xhr.send(post_str({
        "search": search
    }));
}

/**
* Функція рендерингу таблиці
*
* @param {Array} data - дані для рендерингу
* @return void
*/
function render(data) {
    table.innerHTML = '';
    for (let i = 0; i < data.length; i++) {
        let tr = document.createElement('tr');

```

```

        tr.innerHTML = '<th scope="row">' + (i + 1)
+ '</th>'

        tr.innerHTML += '<td>' + sum_strs(' - ',
data[i]['make'], data[i]['model'], data[i]['license_plate']) +
'</td>'

        tr.innerHTML += '<td>' +
data[i]['service_name'] + '</td>'

        tr.innerHTML += '<td>' + '<div class="form-
check form-check-reverse"><input id="checkbox-' + data[i]['id']
+ '" class="form-check-input checkbox" type="checkbox"></div>' +
'</td>'

        table.appendChild(tr);
    }

    for (let i = 0; i < check.length; i++) {
        check[i].addEventListener('change', () => {
            if (check[i].checked) {
                const xhr = new XMLHttpRequest();
                xhr.open('POST',
'/employee/task/complete/' + data[i]['id']);
                xhr.setRequestHeader('X-Requested-
With', 'XMLHttpRequest');

                xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

                xhr.onload = function() {
                    if (xhr.status == 200) {
                        console.log(xhr.response);
                        send();
                    } else {
                        document.getElementById('che
ckbox-' + xhr.responseText).checked = false;
                    }
                };
                xhr.send();
            }
        })
    }

```

```

        });
    }
}

});

/**
 * Функція перетворення об'єкту в строку
 *
 * @param {Object} data - об'єкт для перетворення
 * @return {String} - перетворений об'єкт
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

/**
 * Функція склеювання строк
 *
 * @param {String} split - розділювач
 * @param {...any} theArgs - строки для склеювання
 * @return {String} - склеєна строка
 */
function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

```



```

    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/index/index/action.php

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class indexIndex extends Curswork\ActionClass
{
    /**
     * Контроллер для розподіленню по сторінкам ролей
     *

```

```

* @param $params - параметри запиту
* @param $data - дані для відображення
* @return void
*/
public function action_index($params, $data)
{
    $role = $this->model->get_role($this->session->user_id);
    if ($role) {
        $this->response->redirect('/' . $role .
'/index/index');
    }
}

/**
 * Контроллер для для відображення сторінки 404
 *
 * @param $params - параметри запиту
 * @param $data - дані для відображення
 * @return void
 */
public function action_404($params, $data)
{
    $this->response->header[] = 'HTTP/1.1 404 Not Found';
    $data['title'] = '404 · ' . $this->config->projectName;
    $view = '404';
    $this->view->out($view, $data);
}

/**
 * Контроллер для для відображення сторінки входу
 *
 * @param $params - параметри запиту
 * @param $data - дані для відображення
 * @return void
 */
public function action_login($params, $data)

```

```

{
    $this->response->header[] = 'HTTP/1.1 401 Unauthorized';
    $data['title'] = 'Вхід · ' . $this->config->projectName;
    $view = 'login';
    $this->view->out($view, $data);
}

/**
 * Функція для авторизації користувача
 *
 * @return void
 */
public function auth()
{
    if (!isset($this->request->post->login) || !isset($this->request->post->password)) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if (strlen($this->request->post->login) < 3 ||
    strlen($this->request->post->password) < 3) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    $auth = $this->model->auth_user(urlencode($this->request->post->login), urlencode($this->request->post->password));

    if ($auth) {
        header('HTTP/1.1 200 OK');
    } else {
        header('HTTP/1.1 401 Unauthorized');
    }
}

/**
 * Контроллер для для виходу з аккаунту

```

```

*
* @param $params - параметри запиту
* @param $data - дані для відображення
* @return void
*/
public function action_logout($params, $data)
{
    $this->model->logout();
    unset($_COOKIE[$this->config->sessionKey]);
    setcookie($this->config->sessionKey, null, 0, '/');
    $this->response->redirect('/index/index/login');
}

/**
 * Контроллер для для відображення хедера сайту
 *
 * @param $params - параметри запиту
 * @param $data - дані для відображення
 * @return void
 */
public function action_head($params, $data)
{
    $view = "head";
    $data['account'] = $this->model->get_account();
    $this->view->out($view, $data);
}
}

```

Файл `app/index/index/model.php`

```
<?
```

```

/**
 * License
 *

```

```

* Copyright (c) 2023 Dmytro Zamkovyi
*
* You are prohibited from:
*
* - Using any part of this code for any purpose.
* - Copying or reproducing any part of this code in any form.
* - Modifying or altering any part of this code.
*
* See the full text of the license in the LICENSE file in the
root directory of this repository.
*/

class indexIndexModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для авторизації користувача
     *
     * @param $login - логін користувача
     * @param $password - пароль користувача
     * @return bool
     */
    public function auth_user($login, $password)
    {
        $user_id = $this->db->query('SELECT "id" FROM employee
WHERE login = \'' . $this->db->escape($login) . '\'' and password
= \'' . hash('md5', $this->db->escape($password)) . '\'';');
        if ($user_id) {
            $user_id = (int)$user_id[0]['id'];
            $this->db->query('UPDATE session SET id_employee = '
. $user_id . ' WHERE session_key = \'' . $this->session->session
. '\'';');
            $this->session->load();
            return $this->auth->get_access();
        }
        return false;
    }
}

```

```

/**
 * Метод моделі для виходу користувача
 *
 * @return bool
 */
public function logout()
{
    $this->db->query('DELETE FROM "session" WHERE
session_key = \'' . $this->session->session . '\';');
    return !$this->auth->get_access();
}

/**
 * Метод моделі для отримання ролі користувача
 *
 * @param $user_id - id користувача
 * @return string
 */
public function get_role($user_id)
{
    $res = $this->db->query('SELECT role.role_name as role
FROM employee, role WHERE id_role = role.id AND employee.id = '
. $user_id . ');');
    if ($res) {
        return $res[0]['role'];
    }
    return false;
}

/**
 * Метод моделі для отримання імені користувача
 *
 * @return string
 */
public function get_account()

```

```

        {
            $res = $this->db->query('SELECT first_name, last_name,
fathers_name FROM employee WHERE id=' . $this->session->user_id
. ';' );
            if ($res) {
                return trim(implode(' ', $res[0]));
            }
            return '';
        }
    }
}

```

Файл `app/index/index/view_404.php`

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>

```

```

    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
    <div style="text-align: center;">
        <h1>404</h1>
        <hr>
        <h2>Сторінку не знайдено</h2>
        <a href="/">На головну</a>
        <br>
        <button class="btn btn-primary" style="margin: 10px;"
onclick="window.location =
'/index/index/logout';">Вийти</button>
    </div>
</body>

</html>

```

Файл `app/index/index/view_head.php`

```

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

```

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

<header>

```
<div class="px-3 py-2 text-bg-dark border-bottom"
style="vertical-align: middle;">
  <div class="container">
    <nav class="d-flex flex-wrap align-items-center
justify-content-center justify-content-lg-start">
      <figure class="me-lg-auto">
        <blockquote class="blockquote"
style="margin: 0;">
          <h1><?= $label ?></h1>
        </blockquote>
        <p class="lead" style="margin: 0;"><?=
$account ?></p>
      </figure>
      <ul class="nav col-12 col-lg-auto my-2 justify-
content-center my-md-0 text-small">
        <? for ($i = 0; $i < count($menu); $i++) {
?>
          <li>
            <? if ($menu[$i]['active']) { ?>
              <p class="nav-link text-
secondary" style="margin: 0;">
                <img class="bi d-block mx-
auto mb-1" width="24" height="24" src=<?= $menu[$i]['img']
?>></img>
                <?= $menu[$i]['name'] ?>
              </p>
            <? } else { ?>
              <a href=<?= $menu[$i]['href'] ?>
class="nav-link text-white">
```

```

                                <img class="bi d-block mx-
auto mb-1" width="24" height="24" src=<?= $menu[$i]['img']
?>></img>

                                <?= $menu[$i]['name'] ?>
                                </a>
                                <? } ?>
                                </li>
                                <? } ?>
                                </ul>
                                </nav>
                                </div>
                                </div>
                                </div>
                                </header>

```

Файл app/index/index/view_login.php

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">

```

```

    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body class="d-flex align-items-center py-4 bg-body-tertiary">

    <main class="form-signin w-100 m-auto">
        <h1 class="h3 mb-3 fw-normal m-auto" style="text-align:
center;">Вхід</h1>
        <div class="form-floating">
            <input type="email" class="form-control"
id="floatingInput" placeholder="name@example.com" name="login">
            <label for="floatingInput">Логін</label>
        </div>
        <div class="form-floating">
            <input type="password" class="form-control"
id="floatingPassword" placeholder="Password" name="password">
            <label for="floatingPassword">Пароль</label>
        </div>
        <output hidden id="auth-error" class="text-
danger">Неправильний логін чи пароль</output>
        <output hidden id="log-error" class="text-danger">Логін
не може бути коротше 3-х символів</output>
        <output hidden id="pwd-error" class="text-danger">Пароль
не може бути коротше 3-х символів</output>
        <button id="signin" class="btn btn-primary w-100 py-2"
type="submit">Увійти</button>
    </main>

```

```

<style>
    html,
    body {
        height: 100%;
    }

    .form-signin {
        max-width: 330px;
        padding: 1rem;
    }

    .form-signin .form-floating:focus-within {
        z-index: 2;
    }

    .form-signin input[type="email"] {
        margin-bottom: -1px;
        border-bottom-right-radius: 0;
        border-bottom-left-radius: 0;
    }

    .form-signin input[type="password"] {
        margin-bottom: 10px;
        border-top-left-radius: 0;
        border-top-right-radius: 0;
    }
</style>

<script>
    document.addEventListener("DOMContentLoaded",
function(e) {
        let btn = document.getElementById('signin');
        let login =
document.getElementById('floatingInput');
        let password =
document.getElementById('floatingPassword');

```

```

        let authError = document.getElementById('auth-
error');

        let logError = document.getElementById('log-error');
        let pwdError = document.getElementById('pwd-error');

        /**
         * ajax запит на авторизацію
         *
         * @return void
         */
        btn.addEventListener('click', function(e) {
            btn.disabled = true;
            let error = false;

            let xhr = new XMLHttpRequest();
            xhr.open('POST', '/index/index/index');
            xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
            xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
            xhr.onload = () => {
                if (xhr.status == 200) {
                    window.location.href =
'/index/index/index';
                } else {
                    authError.hidden = false;
                    btn.disabled = false;
                }
            };
            xhr.send('login=' +
encodeURIComponent(login.value) + '&password=' +
encodeURIComponent(password.value));
        });
    });
</script>

```

```

        <!-- <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script> -->
</body>

</html>

```

Файл `app/index/schedule/action.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class indexSchedule extends Curswork\ActionClass
{
    /**
     * Контроллер для виводу розкладу
     *
     * @param $params - параметри запиту
     * @param $data - дані для відображення
     */
}

```

```

public function action_index($params, $data)
{
    $view = "schedule";
    $this->view->out($view, $data);
}

/**
 * Контроллер для отримання розкладу за період часу
 *
 * @param $params - параметри запиту
 * @param $data - дані для відображення
 */
public function action_schedule($params, $data)
{
    header('HTTP/1.1 200 OK');
    header('Content-Type: application/json; charset=utf-8');
    $schedule = $this->model->get_schedule($this->request-
    >post->month, $this->request->post->year);
    echo $schedule;
}
}

```

Файл `app/index/schedule/model.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.

```

```

* - Modifying or altering any part of this code.
*
* See the full text of the license in the LICENSE file in the
root directory of this repository.
*/

```

```

class indexScheduleModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для отримання розкладу користувача
     *
     * @param $month - місяць
     * @param $year - рік
     * @return string
     */
    public function get_schedule($month, $year)
    {
        $schedule = $this->db->query('SELECT schedule_month,
schedule_year, schedule.schedule FROM schedule,
schedule_employee WHERE schedule.id =
schedule_employee.id_schedule AND id_employee = ' . $this-
>session->user_id . ' AND schedule_month = ' . $month . ' AND
schedule_year = ' . $year . ';' );
        if ($schedule) {
            return json_encode($schedule[0]);
        }
        return "{}";
    }
}

```

Файл app/index/schedule/view_schedule.php

```

<!--
License

```


You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

```
<div class="container">
  <p id="error" class="text-danger" hidden>Помилка, неможливо
вивести розклад</p>
  <div class="d-flex flex-wrap align-items-center justify-
content-center justify-content-lg-start">
    <h1 class="me-lg-auto" id="calendar_title">Грудень ·
2023</h1>
    <div>
      <button class="btn">&lt;Попередній</button>
      <button class="btn">Поточний</button>
      <button class="btn">Наступний&gt;</button>
    </div>
  </div>
  <div id="table" class="calendar">
    <div class="calendar_header">
      <p>Понеділок</p>
    </div>
    <div class="calendar_header">
      <p>Вівторок</p>
    </div>
    <div class="calendar_header">
      <p>Середа</p>
    </div>
    <div class="calendar_header">
```

```

        <p>Четвер</p>
    </div>
    <div class="calendar_header">
        <p>П'ятниця</p>
    </div>
    <div class="calendar_header">
        <p>Субота</p>
    </div>
    <div class="calendar_header">
        <p>Неділя</p>
    </div>
</div>
</div>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        render_calendar();

        let shift = 0;

        const xhr = new XMLHttpRequest();
        const div = document.getElementById('table');
        const btn_prev =
document.getElementsByClassName('btn')[0];
        const btn_curr =
document.getElementsByClassName('btn')[1];
        const btn_next =
document.getElementsByClassName('btn')[2];
        const title = document.getElementById('calendar_title');

        /**
         * Функція для переключення на попередній місяць
         *
         * @return void
         */
        btn_prev.addEventListener('click', () => {

```

```

        shift--;
        while (div.children.length > 7) {
            div.removeChild(div.lastChild);
        }
        render_calendar(shift);
        title.innerHTML = date(shift);
        send_request(shift);
    });

/**
 * Функція для переключення на поточний місяць
 *
 * @return void
 */
btn_curr.addEventListener('click', () => {
    shift = 0;
    while (div.children.length > 7) {
        div.removeChild(div.lastChild);
    }
    render_calendar(shift);
    title.innerHTML = date(shift);
    send_request(shift);
});

/**
 * Функція для переключення на наступний місяць
 *
 * @return void
 */
btn_next.addEventListener('click', () => {
    shift++;
    while (div.children.length > 7) {
        div.removeChild(div.lastChild);
    }
    render_calendar(shift);
    title.innerHTML = date(shift);
});

```

```

        send_request(shift);
    });

    send_request();

/**
 * Функція для отримання поточної дати в вигляді місяця
і року
 *
 * @param $shift - здиг місяця
 * @return string
 */
function date(shift) {
    const date = {
        0: 'Січень',
        1: 'Лютий',
        2: 'Березень',
        3: 'Квітень',
        4: 'Травень',
        5: 'Червень',
        6: 'Липень',
        7: 'Серпень',
        8: 'Вересень',
        9: 'Жовтень',
        10: 'Листопад',
        11: 'Грудень'
    };

    const now = new Date();
    now.setMonth(now.getMonth() + shift);
    return date[now.getMonth()] + ' · ' +
now.getFullYear();
}

});

/**

```

```

* Функція для відображення календаря
*
* @param $month - здви́г місяця
* @param $year - здви́г року
* @return void
*/

function render_calendar(month = 0, year = 0) {
    let [table, current_month, current_year] =
create_table_array(month, year);
    let div = document.getElementById('table');

    for (let i = 0; i < table.length; i++) {
        let child = document.createElement('div');
        child.innerHTML = '<p>' + table[i].getDate() +
'</p>';

        child.setAttribute("month", table[i].getMonth());
        child.setAttribute("date", table[i].getDate());
        if (table[i].getMonth() == current_month) {
            child.classList.add("this_month");
        } else {
            child.classList.add("not_this_month");
        }

        if (table[i].getDate() == new Date().getDate() &&
table[i].getMonth() == new Date().getMonth() &&
table[i].getFullYear() == new Date().getFullYear()) {
            child.classList.add("today");
        }

        div.appendChild(child);
    }

/**
* Функція для створення масиву дат для відображення
*
* @param $month - зди́г місяця
* @param $year - здви́г року
* @return array

```

```

    */
    function create_table_array(month = 0, year = 0) {
        let [current_year, current_month] =
get_current_date(month, year);

        let date = new Date(current_year, current_month);
        let table = Array();
        while (true) {
            if (date.getDay() == 1) {
                break;
            }
            date.setDate(date.getDate() - 1);
        }

        while (true) {
            if (date.getDay() == 1) {
                if (current_month == 11) {
                    if (date.getMonth() == 0) {
                        break;
                    }
                } else {
                    if (current_month == 0) {
                        if (date.getMonth() == 1) {
                            break;
                        }
                    } else {
                        if (date.getMonth() > current_month)
{
                            break;
                        }
                    }
                }
            }
        }

        table.push(new Date(date.getTime()));
        date.setDate(date.getDate() + 1);
    }

```

```

    }

    return [table, current_month, current_year];
}
}

/**
 * Функція для отримання дати
 *
 * @param $month - здиг місяця
 * @param $year - здвиг року
 * @return array
 */
function get_current_date(month = 0, year = 0) {
    let now = new Date();
    while (month >= 12) {
        month -= 12;
        year++;
    }
    while (month <= -12) {
        month += 12;
        year--;
    }
    let current_month = now.getMonth() + month;
    if (current_month > 11) {
        current_month -= 12;
        year++;
    }
    if (current_month < 0) {
        current_month += 12;
        year--;
    }
    let current_year = now.getFullYear() + year;
    return [current_year, current_month];
}

```

```

/**
 * Функція для відправки запиту на отримання розкладу
 *
 * @param $month - здиг місяця
 * @param $year - здвиг року
 * @return void
 */
function send(month = 0, year = 0) {
    let [current_year, current_month] =
get_current_date(month, year);

    const xhr = new XMLHttpRequest();
    xhr.open('POST', '/index/schedule/schedule');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader('Content-type', 'application/x-www-
form-urlencoded');
    xhr.onload = () => {
        if (xhr.status !== 200) {
            document.getElementById('error').hidden = false;
            return;
        }

        document.getElementById('error').hidden = true;

        if (xhr.responseText == '{}') {
            return;
        }

        const div = document.getElementById('table');

        let res = JSON.parse(xhr.responseText);
        res['schedule'] = JSON.parse(res['schedule']);
        for (let i = 7; i < div.children.length; i++) {

```



```

        if (div.children[i].getAttribute('month') ==
res['schedule_month'] &&
res['schedule'][div.children[i].getAttribute('date')]) {
            if
(!res['schedule'][div.children[i].getAttribute('date')]['weekend
']) {
                div.children[i].innerHTML += '<p
class="schedule" align="center">' +
res['schedule'][div.children[i].getAttribute('date')]['start'] +
' - ' +
res['schedule'][div.children[i].getAttribute('date')]['end'] +
'</p>';
            }
        }
    }
};
xhr.send('month=' + encodeURIComponent(current_month) +
'&year=' + encodeURIComponent(current_year));
}

/**
 * Функція для відправки запиту на отримання розкладу для
відображення в календарі
 *
 * @param $month - здинг місяця
 * @param $year - здинг року
 * @return void
 */
function send_request(month = 0, year = 0) {
    send(month - 1, year);
    send(month, year);
    send(month + 1, year);
}
</script>

<style>

```

```

.calendar {
    display: grid;
    grid-template-columns: repeat(7, 1fr);
}

.calendar div {
    border: 1px solid blanchedalmond;
    height: 100%;
    min-height: 100px;
}

.this_month p {
    color: white;
}

.not_this_month p {
    color: gray;
}

.today {
    background-color: cadetblue;
}

.today p {
    color: black;
}

.calendar div p {
    margin: 0 6px;
    padding: 0;
    font-size: 1.5em;
}

.calendar_header {
    background-color: cadetblue;
    text-align: center;
}

```

```

        height: 100% !important;
        min-height: 0 !important;
    }

    .calendar_header p {
        margin: 0 !important;
        padding: 0 !important;
        font-size: 1.5em !important;
        color: black;
    }

    .btn {
        border: 2px solid white;
        border-radius: 5px;
        width: 150px;
    }

    .btn:active {
        background-color: white !important;
        color: black !important;
    }

    .not_this_month .schedule {
        color: grey !important;
    }

    .this_month .schedule {
        background-color: #343438;
    }

    .today .schedule {
        background-color: cadetblue;
    }
</style>

```

Файл app/manager/index/action.php

<?

```
/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */
```

```
class managerIndex extends Curswork\ActionClass
{
    /**
     * Контроллер для перенаправления на головну сторінку
менеджера
     *
     * @param $params - параметри
     * @param $data - дані
     */
    public function action_index($params, $data)
    {
        $this->response->redirect('/manager/order/index');
    }

    /**
     * Контроллер для відображення списку нових замовлень
     *

```

```

* @param $params - параметри
* @param $data - дані
*/
public function action_schedule($params, $data)
{
    $view = "schedule";

    $data['label'] = 'Мій графік';
    $data['menu'] = array(
        array(
            'name' => 'Замовлень',
            'href' => '/manager/order/index',
            'active' => false,
            'img' => '/data/icon/order.png'
        ),
        array(
            'name' => 'Розклад',
            'href' => '/manager/index/schedule',
            'active' => true,
            'img' => '/data/icon/calendar_activ.png'
        ),
        array(
            'name' => 'Вихід',
            'href' => '/index/index/logout',
            'active' => false,
            'img' => '/data/icon/logout.png'
        ),
    );
    $data['head'] = $this->load('index', 'index', 'head',
'', $data);

    $data['title'] = 'Розклад · ' . $this->config-
>projectName;
    $data['schedule'] = $this->load('index', 'schedule',
'index');

```

```

        $this->view->out($view, $data);
    }
}

```

Файл `app/manager/index/view_schedule.php`

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcdSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">

```

```

</head>

<body>
    <?= $head ?>
    <?= $schedule ?>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл app/manager/order/action.php

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class managerOrder extends Curswork\ActionClass
{
    /**

```

```

* Контроллер для відображення списку замовлень (головної
сторінки менеджера/замовлення)
*
* @param string $params - параметри
* @param array $data - дані
* @return void
*/
public function action_index($params, $data)
{
    $view = "index";
    $data['label'] = 'Замовлення';
    $data['menu'] = array(
        array(
            'name' => 'Замовлення',
            'href' => '/manager/order/index',
            'active' => true,
            'img' => '/data/icon/order_activ.png'
        ),
        array(
            'name' => 'Розклад',
            'href' => '/manager/index/schedule',
            'active' => false,
            'img' => '/data/icon/calendar.png'
        ),
        array(
            'name' => 'Вихід',
            'href' => '/index/index/logout',
            'active' => false,
            'img' => '/data/icon/logout.png'
        ),
    );
    $data['head'] = $this->load('index', 'index', 'head',
'', $data);
    $data['title'] = 'Замовлення · ' . $this->config-
>projectName;

```



```

        $data['success'] = $this->session->success == null ?
'hidden' : '';
        unset($this->session->success);

        $this->view->out($view, $data);
    }

    /**
     * Контроллер для пошуку замовлень
     *
     * @param string $params - параметри
     * @param array $data - дані
     * @return void
     */
    public function action_search($params, $data)
    {
        $data['first_name'] = isset($this->request->post-
>first_name) ? urldecode($this->request->post->first_name) :
null;

        $data['last_name'] = isset($this->request->post-
>last_name) ? urldecode($this->request->post->last_name) : null;

        $data['father_name'] = isset($this->request->post-
>father_name) ? urldecode($this->request->post->father_name) :
null;

        $data['phone_number'] = isset($this->request->post-
>phone_number) ? urldecode($this->request->post->phone_number) :
null;

        $data['car'] = isset($this->request->post->car) ?
urldecode($this->request->post->car) : null;

        $data['start_date'] = isset($this->request->post-
>start_date) ? urldecode($this->request->post->start_date) :
null;

        $data['end_date'] = isset($this->request->post-
>end_date) ? urldecode($this->request->post->end_date) : null;

        $data['complete'] = isset($this->request->post-
>complete) ? urldecode($this->request->post->complete) : null;
    }

```

```

        $data['pay'] = isset($this->request->post->pay) ?
urldecode($this->request->post->pay) : null;

        if ($data['first_name'] == null) {
            unset($data['first_name']);
        } elseif (strlen($data['first_name']) < 3 ||
strlen($data['first_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['last_name'] == null) {
            unset($data['last_name']);
        } elseif (strlen($data['last_name']) < 3 ||
strlen($data['last_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['father_name'] == null) {
            unset($data['father_name']);
        } elseif (strlen($data['father_name']) < 3 ||
strlen($data['father_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if ($data['phone_number'] == null) {
            unset($data['phone_number']);
        } else {
            preg_match_all('/^[ ]*\+?([0-9]{2})[-\(\ ]*([0-
9]{1})[-\(\ ]*([0-9]{2})[-\(\ ]*([0-9]{3})[-\ ]*([0-9]{2})[-\ ]*([0-
9]{2})[ ]*$/ ', $data['phone_number'], $matches);
            if (count($matches[0]) == 0) {
                header('HTTP/1.1 400 Bad Request');
                return;
            }
        }
    }
}

```

```

    }
    $matches = array_slice($matches, 1);
    for ($i = 0; $i < count($matches); $i++) {
        $matches[$i] = (int)$matches[$i][0];
    }
    $data['phone_number'] = (int)implode($matches);
}

if ($data['car'] == null) {
    unset($data['car']);
} elseif (strlen($data['car']) == 0 ||
strlen($data['car']) > 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if ($data['start_date'] == null) {
    unset($data['start_date']);
} elseif (strlen($data['start_date']) != 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if ($data['end_date'] == null) {
    unset($data['end_date']);
} elseif (strlen($data['end_date']) != 10) {
    header('HTTP/1.1 400 Bad Request');
    return;
}

if (isset($data['start_date']) &&
isset($data['end_date'])) {
    if (strtotime($data['start_date']) >
strtotime($data['end_date'])) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
}

```

```

        }
    }

    if ($data['complete'] == null) {
        unset($data['complete']);
    } elseif ($data['complete'] != 'true' &&
$data['complete'] != 'false') {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if ($data['pay'] == null) {
        unset($data['pay']);
    } elseif ($data['pay'] != 'true' && $data['pay'] !=
'false') {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    echo json_encode($this->model->search($data));
}

/**
 * Контроллер для редагування замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_update($params, $data)
{
    $data['order_id'] = $params;
    $data['discount'] = (int)urldecode($this->request->post-
>discount);

    if (!$this->model->update($data)) {

```

```

        header('HTTP/1.1 400 Bad Request');
        return;
    }
    header('HTTP/1.1 200 Ok');
}

/**
 * Контроллер для видалення замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_delete($params, $data)
{
    $data['order_id'] = (int)$params;
    $this->model->delete($data['order_id']);
    $this->response->redirect('/manager/order/index');
}

/**
 * Контроллер для створення замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_add($params, $data)
{
    $this->session->success = true;
}

/**
 * Контроллер для перегляду сторінки створення замовлення
 *
 * @param string $params - параметри

```

```

    * @param array $data - дані
    * @return void
    */
public function action_create($params, $data)
{
    $view = "create";

    $data['title'] = 'Створення замовлення · ' . $this->config->projectName;

    $data['sto_list'] = $this->model->get_sto_list();

    $this->view->out($view, $data);
}

/**
 * Контроллер для перегляду сторінки редагування замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_edit($params, $data)
{
    $data['order_id'] = $params;
    $view = "edit";

    $data['order_info'] = $this->model->get_order($data['order_id']);
    if (!$data['order_info']) {
        header('HTTP/1.1 404 Not Found');
        $this->response->redirect('/index/index/404');
        return;
    }
}

```

```

        $data['task_list'] = $this->model-
>get_task($data['order_id']);
        if (!$data['task_list']) {
            header('HTTP/1.1 404 Not Found');
            $this->response->redirect('/index/index/404');
            return;
        }

        $data['sto_info'] = $this->model-
>get_sto($data['task_list'][0]['employee_id']);

        $data['title'] = 'Редагування замовлення · ' . $this-
>config->projectName;

        $this->view->out($view, $data);
    }

/**
 * Контроллер для отримання списку завдань замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_task($params, $data)
{
    $data['order_id'] = $params;

    $data['task_list'] = $this->model-
>get_task($data['order_id']);
    if (!$data['task_list']) {
        header('HTTP/1.1 404 Not Found');
        $this->response->redirect('/index/index/404');
        return;
    }

```

```

        echo json_encode($data['task_list']);
    }

/**
 * Контроллер для видалення завдання замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_delete_task($params, $data)
{
    $data['task_id'] = $params;

    if (!$this->model->delete_task($data['task_id'])) {
        header('HTTP/1.1 400 Bad Request');
    } else {
        header('HTTP/1.1 200 Ok');
    }
}

/**
 * Контроллер для отримання списку працівників СТО
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_employee_list($params, $data)
{
    $data['sto_name'] = urldecode($this->request->post->sto_name);

    $result = $this->model->get_employee_list($data['sto_name']);
    if (!$result) {

```



```

        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

/**
 * Контроллер для отримання списку послуг
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_get_task_list($params, $data)
{
    $result = $this->model->get_task_list();
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

/**
 * Контроллер для додавання завдання замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void
 */
public function action_add_task($params, $data)
{
    $data['order_id'] = urldecode($params);
    $data['name'] = explode(' ', urldecode($this->request-
>post->name));
    $data['task'] = urldecode($this->request->post->task);

```

```

        if (!$this->model->add_task($data)) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        header('HTTP/1.1 200 Ok');
    }

/**
 * Контроллер для пошуку клієнтів
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з клієнтами
 */
public function action_search_client($params, $data)
{
    $data['search'] = urldecode($this->request->post->search);

    $result = $this->model->search_client($data['search']);
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

/**
 * Контроллер для додавання клієнта
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з клієнтом
 */
public function action_add_client($params, $data)

```

```

    {
        $data['first_name'] = urldecode($this->request->post->first_name);
        $data['last_name'] = urldecode($this->request->post->last_name);
        $data['fathers_name'] = $this->request->post->father_name;
        $data['phone_number'] = urldecode($this->request->post->phone_number);

        $search = $this->model->search_client_by_phone($data['phone_number']);

        if ($search) {
            header('HTTP/1.1 200 Ok');
            echo json_encode($search);
            return;
        }

        if (strlen($data['first_name']) < 3 ||
            strlen($data['first_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        if (strlen($data['last_name']) < 3 ||
            strlen($data['last_name']) > 40) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }

        preg_match_all('/^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/', $data['phone_number'], $matches);
        if (count($matches[0]) == 0) {
            header('HTTP/1.1 400 Bad Request');
        }
    }

```

```

        return;
    }
    $matches = array_slice($matches, 1);
    for ($i = 0; $i < count($matches); $i++) {
        $matches[$i] = (int)$matches[$i][0];
    }
    $data['phone_number'] = (int)implode($matches);

    $add = $this->model->add_client($data);
    if (!$add) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($this->model->
>search_client_by_phone($data['phone_number']));
    }

/**
 * Контроллер для пошуку автомобілів
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з автомобілями
 */
public function action_search_car($params, $data)
{
    $data['search'] = urldecode($this->request->post->
>search);

    $result = $this->model->search_car($data['search']);
    if (!$result) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }
    echo json_encode($result);
}

```

```

/**
 * Контроллер для додавання автомобіля
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з автомобілем
 */
public function action_add_car($params, $data)
{
    $data['make'] = urldecode($this->request->post->make);
    $data['model'] = urldecode($this->request->post->model);
    $data['year'] = urldecode($this->request->post->car_year);
    $data['license_plate'] = urldecode($this->request->post->license_plate);
    $data['description'] = urldecode($this->request->post->description);

    $search = $this->model->search_car($data['license_plate']);

    if ($search) {
        header('HTTP/1.1 200 Ok');
        echo json_encode($search);
        return;
    }

    if (strlen($data['make']) < 3 || strlen($data['make']) >
40) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if (strlen($data['model']) < 3 || strlen($data['model'])
> 40) {

```

```

        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if (strlen($data['year']) != 4) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    if (strlen($data['license_plate']) < 3 ||
strlen($data['license_plate']) > 10) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    $add = $this->model->add_car($data);

    if (!$add) {
        header('HTTP/1.1 400 Bad Request');
        return;
    }

    echo json_encode($this->model->search_car($data['license_plate']));
}

/**
 * Контроллер для створення замовлення
 *
 * @param string $params - параметри
 * @param array $data - дані
 * @return void - масив з замовленням
 */
public function action_create_order($params, $data)
{
    $data['client_id'] = urldecode($this->request->post->client_id);

```

```

        $data['car_id'] = urldecode($this->request->post->car_id);

        $result = $this->model->create_order($data);
        if (!$result) {
            header('HTTP/1.1 400 Bad Request');
            return;
        }
        echo json_encode($result);
    }
}

```

Файл `app/manager/order/model.php`

```

<?

/**
 * License
 *
 * Copyright (c) 2023 Dmytro Zamkovyi
 *
 * You are prohibited from:
 *
 * - Using any part of this code for any purpose.
 * - Copying or reproducing any part of this code in any form.
 * - Modifying or altering any part of this code.
 *
 * See the full text of the license in the LICENSE file in the
root directory of this repository.
 */

class managerOrderModel extends Curswork\ActionClassModel
{
    /**
     * Метод моделі для отримання списку замовлень

```

```

*
* @param $data - масив з даними для пошуку
* @return array - масив з замовленнями
*/
public function search($data)
{
    $sql = 'SELECT "order".id, client.first_name,
client.last_name, client.fathers_name, client.phone_number,
car.make, car.model, car.license_plate, "order".create_date,
"order".completed, "order".payment FROM "order", client, car
WHERE "order".id_client = client.id AND "order".id_car =
car.id';

    if (isset($data['first_name'])) {
        $sql .= ' AND client.first_name LIKE \'%' . $this-
>db->escape($data['first_name']) . '%\'';
    }

    if (isset($data['last_name'])) {
        $sql .= ' AND client.last_name LIKE \'%' . $this-
>db->escape($data['last_name']) . '%\'';
    }

    if (isset($data['father_name'])) {
        $sql .= ' AND client.fathers_name LIKE \'%' . $this-
>db->escape($data['father_name']) . '%\'';
    }

    if (isset($data['phone_number'])) {
        $sql .= ' AND client.phone_number = ' . $this->db-
>escape($data['phone_number']);
    }

    if (isset($data['car'])) {
        $sql .= ' AND car.license_plate LIKE \'%' . $this-
>db->escape($data['car']) . '%\'';
    }

    if (isset($data['start_date'])) {

```



```

        $sql .= ' AND "order".create_date >= \'' . $this->db->escape($data['start_date']) . '\';
    }
    if (isset($data['end_date'])) {
        $sql .= ' AND "order".create_date <= \'' . $this->db->escape($data['end_date']) . '\';
    }
    if (isset($data['complete'])) {
        $sql .= ' AND "order".completed = ' . $this->db->escape($data['complete']);
    }
    if (isset($data['pay'])) {
        $sql .= ' AND "order".payment = ' . $this->db->escape($data['pay']);
    }

    $sql .= ';';

    $result = $this->db->query($sql);

    if ($this->db->last_error) {
        return array();
    }

    for ($i = 0; $i < count($result); $i++) {
        $result[$i]['completed'] = $result[$i]['completed']
== 't';
        $result[$i]['payment'] = $result[$i]['payment'] ==
't';
    }
    return $result;
}

/**
 * Метод моделі для отримання деталей замовлення
 *

```

```

* @param $id - id замовлення
* @return bool|array - масив з деталями замовлення
*/
public function get_order($id)
{
    $sql = 'SELECT "order".create_date, "order".completed,
"order".payment, "order".discount AS order_discount, car.make,
car.model, car.license_plate, client.first_name,
client.last_name, client.fathers_name, client.phone_number,
client.discount FROM "order", car, client WHERE "order".id_car =
car.id AND "order".id_client = client.id AND "order".id = ' .
$this->db->escape($id) . ' ';
    $order_result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($order_result) == 0) {
        return false;
    }
    return $order_result[0];
}

/**
* Метод моделі для отримання списку завдань по замовленню
*
* @param $id - id замовлення
* @return bool|array - масив з завданнями
*/
public function get_task($id)
{
    $sql = 'SELECT task.id, task.is_done, employee.id AS
employee_id, employee.first_name, employee.last_name,
employee.fathers_name, service.service_name, service.price FROM
task, employee, service WHERE task.id_employee = employee.id AND
task.id_service = service.id AND task.id_order = ' . $this->db->
escape($id) . ' ';

```

```

        $tasks_result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return $tasks_result;
    }

    /**
     * Метод моделі для отримання списку СТО
     *
     * @param $employee_id - id працівника
     * @return bool|array - масив з СТО
     */
    public function get_sto($employee_id)
    {
        $sql = 'SELECT sto.id, sto.sto_name FROM employee, sto
        WHERE employee.id_sto = sto.id AND employee.id = ' . $this->db-
        >escape($employee_id) . ' ';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return $result[0];
    }

    /**
     * Метод моделі для видалення завдання по замовленню
     *
     * @param $task_id - id завдання
     * @return bool - результат видалення
     */
    public function delete_task($task_id)
    {
        $sql = 'DELETE FROM task WHERE id = ' . $this->db-
        >escape($task_id) . ' ';
        $this->db->query($sql);
    }

```

```

        if ($this->db->last_error) {
            return false;
        }
        return true;
    }

/**
 * Метод моделі для отримання списку працівників СТО
 *
 * @param $sto_name - назва СТО
 * @return bool|array - масив з працівниками
 */
public function get_employee_list($sto_name)
{
    $sql = 'SELECT employee.id, employee.first_name,
employee.last_name, employee.fathers_name FROM employee, sto
WHERE employee.id_sto = sto.id AND employee.id_role = 3 AND
sto.sto_name = \'' . $this->db->escape($sto_name) . '\';';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return $result;
}

/**
 * Метод моделі для отримання списку послуг
 *
 * @return bool|array - масив з послугами
 */
public function get_task_list()
{
    $sql = 'SELECT service.service_name FROM service;';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }

```

```

    }
    return $result;
}

/**
 * Метод моделі для додавання завдання по замовленню
 *
 * @param $data - масив з даними для додавання
 * @return bool - результат додавання
 */
public function add_task($data)
{
    $sql = 'SELECT id FROM employee WHERE first_name = \'' .
$this->db->escape($data['name'][0]) . '\'';
    if (isset($data['name'][1])) {
        $sql .= ' AND last_name = \'' . $this->db-
>escape($data['name'][1]) . '\'';
    }
    if (isset($data['name'][2])) {
        $sql .= ' AND fathers_name = \'' . $this->db-
>escape($data['name'][2]) . '\'';
    }
    $id_employee = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($id_employee) == 0) {
        return false;
    }
    $id_employee = $id_employee[0]['id'];

    $sql = 'SELECT id FROM service WHERE service_name = \''
. $this->db->escape($data['task']) . '\'';
    $id_service = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }

```

```

    }
    if (count($id_service) == 0) {
        return false;
    }
    $id_service = $id_service[0]['id'];

    $sql = 'INSERT INTO task (id_order, id_service,
id_employee, is_done) VALUES ('. $this->db-
>escape($data['order_id']) . ', ' . $id_service . ', ' .
$id_employee . ', false);';
    $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return true;
}

/**
 * Метод моделі для зміни деталей замовлення
 *
 * @param $data - масив з даними для зміни
 * @return bool - результат зміни
 */
public function update($data)
{
    if ($data['discount'] < 0 || $data['discount'] > 100) {
        return false;
    }

    if ($data['discount'] == 0) {
        $data['discount'] = 'NULL';
    }

    $sql = 'UPDATE "order" SET discount = '. $this->db-
>escape($data['discount']) . ' WHERE id = '. $this->db-
>escape($data['order_id']) . ';';

```

```

        $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        return true;
    }

    /**
     * Метод моделі для втдалення замовлення
     *
     * @param $id - id замовлення
     * @return void
     */
    public function delete($id)
    {
        $sql = 'DELETE FROM task WHERE id_order = ' . $this->db->escape($id) . ' ';
        $this->db->query($sql);

        $sql = 'DELETE FROM "order" WHERE id = ' . $this->db->escape($id) . ' ';
        $this->db->query($sql);
    }

    /**
     * Метод моделі для отримання списку СТО
     *
     * @return bool|array - масив з СТО
     */
    public function get_sto_list()
    {
        $sql = 'SELECT id, sto_name FROM sto;';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {

```

```

        return false;
    }
    return $result;
}

/**
 * Метод моделі для отримання списку послуг
 *
 * @param $search - дані для пошуку
 * @return bool|array - масив з послугами
 */
public function search_client($search)
{
    $sql = 'SELECT * FROM client WHERE first_name LIKE \'%'.
$this->db->escape($search) . '%\' OR last_name LIKE \'%'. $this-
>db->escape($search) . '%\' OR fathers_name LIKE \'%'. $this-
>db->escape($search) . '%\';';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($result) != 1) {
        return false;
    }
    return $result;
}

/**
 * Метод моделі для пошуку клієнта по номеру телефона
 *
 * @param $phone - номер телефона
 * @return bool|array - масив інформації про клієнта
 */
public function search_client_by_phone($phone)
{

```



```

        $sql = 'SELECT * FROM client WHERE phone_number = ' .
$this->db->escape($phone) . ' ';
        $result = $this->db->query($sql);
        if ($this->db->last_error) {
            return false;
        }
        if (count($result) != 1) {
            return false;
        }
        return $result;
    }

/**
 * Метод моделі для додавання клієнта
 *
 * @param $data - масив з даними для додавання
 * @return bool - результат додавання
 */
public function add_client($data)
{
    if (!isset($data['fathers_name'])) {
        $sql = 'INSERT INTO client (first_name, last_name,
phone_number) VALUES (\'' . $this->db-
>escape($data['first_name']) . '\', \'' . $this->db-
>escape($data['last_name']) . '\', ' . $this->db-
>escape($data['phone_number']) . ');';
    } else {
        $sql = 'INSERT INTO client (first_name, last_name,
fathers_name, phone_number) VALUES (\'' . $this->db-
>escape($data['first_name']) . '\', \'' . $this->db-
>escape($data['last_name']) . '\', \'' . $this->db-
>escape((string)$data['fathers_name']) . '\', ' . $this->db-
>escape($data['phone_number']) . ');';
    }
    $this->db->query($sql);
    if ($this->db->last_error) {

```

```

        return false;
    }
    return true;
}

/**
 * Метод моделі для пошуку автомобіля по номеру
 *
 * @param $search - номер автомобіля
 * @return bool|array - масив з інформацією про автомобіль
 */
public function search_car($search)
{
    $sql = 'SELECT * FROM car WHERE license_plate = \'' .
$this->db->escape($search) . '\';';
    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    if (count($result) != 1) {
        return false;
    }
    return $result;
}

/**
 * Метод моделі для додавання автомобіля
 *
 * @param $data - масив з даними для додавання
 * @return bool - результат додавання
 */
public function add_car($data)
{
    if (!isset($data['year']) &&
!isset($data['description'])) {

```

```

        $sql = 'INSERT INTO car (make, model, license_plate)
VALUES (\'' . $this->db->escape($data['make']) . '\', \'' .
$this->db->escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\');';
    } else {
        if (!isset($data['year'])) {
            $sql = 'INSERT INTO car (make, model,
license_plate) VALUES (\'' . $this->db->escape($data['make']) .
'\', \'' . $this->db->escape($data['model']) . '\', \'' . $this-
>db->escape($data['license_plate']) . '\');';
        } else {
            $sql = 'INSERT INTO car (make, model,
license_plate, car_year) VALUES (\'' . $this->db-
>escape($data['make']) . '\', \'' . $this->db-
>escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\', ' . $this->db-
>escape($data['year']) . ');';
        }

        if (!isset($data['description'])) {
            $sql = 'INSERT INTO car (make, model,
license_plate) VALUES (\'' . $this->db->escape($data['make']) .
'\', \'' . $this->db->escape($data['model']) . '\', \'' . $this-
>db->escape($data['license_plate']) . '\');';
        } else {
            $sql = 'INSERT INTO car (make, model,
license_plate, description) VALUES (\'' . $this->db-
>escape($data['make']) . '\', \'' . $this->db-
>escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\', \'' . $this->db-
>escape($data['description']) . '\');';
        }
    }

    if (isset($data['year']) && isset($data['description']))
{

```

```

        $sql = 'INSERT INTO car (make, model, license_plate,
car_year, description) VALUES (\'' . $this->db-
>escape($data['make']) . '\', \'' . $this->db-
>escape($data['model']) . '\', \'' . $this->db-
>escape($data['license_plate']) . '\', ' . $this->db-
>escape($data['year']) . ', \'' . $this->db-
>escape($data['description']) . '\');';
    }

    $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return true;
}

/**
 * Метод моделі для отримання списку автомобілів
 *
 * @param $search - дані для пошуку
 * @return bool|array - масив з автомобілями
 */
public function create_order($data)
{
    $sql = 'INSERT INTO "order" (id_client, id_car,
create_date, completed, payment) VALUES (' . $this->db-
>escape($data['client_id']) . ', ' . $this->db-
>escape($data['car_id']) . ', now(), false, false) RETURNING
id;';

    $result = $this->db->query($sql);
    if ($this->db->last_error) {
        return false;
    }
    return $result;
}
}

```

Файл `app/manager/order/view_create.php`

```
<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
```

```

<header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">
    <a id="back" href="/manager/order/index" class="d-flex
align-items-center mb-3 mb-md-0 me-md-auto link-body-emphasis
text-decoration-none" style="margin-left: 16px;">
        
        <span class="fs-4">Назад</span>
    </a>

    <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;" hidden>Зберегти</button>
</header>
<main>
    <div class="container">
        <div class="input-group mb-3">
            <span class="input-group-text">Замовник</span>
            <input id="client" type="text" class="form-
control" disabled>
            <button id="add-client-btn" class="btn btn-
outline-primary" type="button" data-bs-toggle="modal" data-bs-
target="#add-client-model">Вибрати</button>
        </div>

        <div id="car-div" class="input-group mb-3" hidden>
            <span class="input-group-text">Автомобіль</span>
            <input id="car" type="text" class="form-control"
disabled>
            <button id="add-car-btn" class="btn btn-outline-
primary" type="button" data-bs-toggle="modal" data-bs-
target="#add-car-model">Вибрати</button>
        </div>

        <div id="sto-list-div" class="input-group mb-3"
hidden>
            <span class="input-group-text">СТО</span>

```

```

        <select id="sto-list" class="form-select"
autocomplete="off">
            <option selected hidden value=""></option>
            <? foreach ($sto_list as $item) : ?>
                <option value="<?= $item['sto_name']
?>"><?= $item['sto_name'] ?></option>
            <? endforeach; ?>
        </select>
    </div>

    <p id="error" class="text-danger" hidden></p>
</div>

<div id="new-task-div" class="container" hidden>
    <button id="new-task" type="button" class="btn btn-
primary" data-bs-toggle="modal" data-bs-target="#add-task-
model">Нове завдання</button>
</div>

<div id="table-div" style="margin-left: 16px; margin-
right: 16px;" hidden>
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Назва послуги</th>
                <th scope="col">Виконавець</th>
                <th scope="col">Ціна</th>
                <th scope="col">Виконано</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody id="table-body"></tbody>
    </table>
</div>

```

```

        <div class="modal fade" id="add-client-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
            <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
                <div class="modal-content">
                    <div class="modal-header">
                        <h1 class="modal-title fs-5"
id="staticBackdropLabel">Додавання клієнта</h1>
                        <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
                    </div>
                    <div class="modal-body">
                        <div class="input-group mb-3">
                            <input id="client-search"
type="text" class="form-control" placeholder="Пошук">
                            <button id="client-search-btn"
class="btn btn-outline-primary" type="button">Знайти</button>
                        </div>

                        <div class="input-group mb-3">
                            <span class="input-group-
text">ПІБ</span>
                            <input id="first-name" type="text"
class="form-control" placeholder="Прізвище">
                            <input id="last-name" type="text"
class="form-control" placeholder="Ім'я">
                            <input id="fathers-name" type="text"
class="form-control" placeholder="Патронім">
                        </div>

                        <div class="input-group mb-3">
                            <span class="input-group-text">Номер
телефону</span>
                            <input id="phone-number" type="text"
class="form-control">

```



```

        <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
    </div>
</div>
<div class="modal-footer">
    <output hidden id="client-error"
class="text-danger"></output>
    <button id="add-client" type="button"
class="btn btn-primary">Додати</button>
</div>
</div>
</div>
</div>

<div class="modal fade" id="add-car-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="staticBackdropLabel">Додавання автомобіля</h1>
                <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="input-group mb-3">
                    <input id="car-search" type="text"
class="form-control" placeholder="Пошук">
                    <button id="car-search-btn"
class="btn btn-outline-primary" type="button">Знайти</button>
                </div>

                <div class="input-group mb-3">

```

```

        <span class="input-group-
text">Мапка</span>

        <input id="make" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Модель</span>

        <input id="model" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Рік
випуску</span>

        <input id="year" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Номерний знак</span>

        <input id="license-plate"
type="text" class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Опис
(опціонально)</span>

        <textarea id="description"
class="form-control" aria-label="With textarea"></textarea>
    </div>
</div>
<div class="modal-footer">

```

```

        <output hidden id="car-error"
class="text-danger"></output>
        <button id="add-car" type="button"
class="btn btn-primary">Додати</button>
    </div>
</div>
</div>
</div>

<div class="modal fade" id="add-task-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту клієнта</h1>
                <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="input-group mb-3">
                    <span class="input-group-
text">Працівник</span>
                    <input id="employee" type="text"
class="form-control" list="employee-list">
                    <datalist id="employee-
list"></datalist>
                </div>

                <div class="input-group mb-3">
                    <span class="input-group-
text">Завдання</span>
                    <input id="task" type="text"
class="form-control" list="task-list">

```

```

        <datalist id="task-list"></datalist>
    </div>
</div>
<div class="modal-footer">
    <output hidden id="error" class="text-
danger"></output>
    <button id="create" type="button"
class="btn btn-primary">Створити</button>
</div>
</div>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const task_modal = new
bootstrap.Modal(document.getElementById('add-task-model'));
        const client_modal = new
bootstrap.Modal(document.getElementById('add-client-model'));
        const car_modal = new
bootstrap.Modal(document.getElementById('add-car-model'));

        const inp_client =
document.getElementById('client');
        const inp_car = document.getElementById('car');
        const inp_sto_list = document.getElementById('sto-
list');

        const error = document.getElementById('error');
        const btn_new_task = document.getElementById('new-
task');

        const table_div = document.getElementById('table-
div');

        const new_task_div = document.getElementById('new-
task-div');

```

```

        const car_div = document.getElementById('car-div');
        const sto_list_div = document.getElementById('sto-
list-div');

        const inp_client_search =
document.getElementById('client-search');
        const btn_client_search =
document.getElementById('client-search-btn');
        const client_error =
document.getElementById('client-error');
        const btn_add_client = document.getElementById('add-
client');
        const inp_client_first_name =
document.getElementById('first-name');
        const inp_client_last_name =
document.getElementById('last-name');
        const inp_client_fathers_name =
document.getElementById('fathers-name');
        const inp_client_phone_number =
document.getElementById('phone-number');

        const inp_car_search = document.getElementById('car-
search');
        const btn_car_search = document.getElementById('car-
search-btn');
        const car_error = document.getElementById('car-
error');
        const btn_add_car = document.getElementById('add-
car');
        const inp_car_make =
document.getElementById('make');
        const inp_car_model =
document.getElementById('model');
        const inp_car_year =
document.getElementById('year');

```

```

        const inp_car_license_plate =
document.getElementById('license-plate');
        const inp_car_description =
document.getElementById('description');

        const btn_client = document.getElementById('add-
client-btn');
        const btn_car = document.getElementById('add-car-
btn');

        const employee_list =
document.getElementById('employee-list');
        const task_list = document.getElementById('task-
list');

        const btn_create =
document.getElementById('create');
        const inp_employee =
document.getElementById('employee')
        const inp_task = document.getElementById('task');

        const btn_save = document.getElementById('save');

        let client_id = null;
        let car_id = null;
        let order_id = null;

        inp_client.value = '';
        inp_car.value = '';
        task_list.value = '';

        /**
        * Функція для очистки полів модального вікна
        *
        * @returns {void}
        */

```

```

    btn_client.addEventListener('click', () => {
        inp_client_search.value = '';
        inp_client_first_name.value = '';
        inp_client_last_name.value = '';
        inp_client_fathers_name.value = '';
        inp_client_phone_number.value = '';
    });

/**
 * Функція для пошуку клієнта
 *
 * @returns {void}
 */
    btn_client_search.addEventListener('click', () => {
        let xhr = new XMLHttpRequest();
        xhr.open('POST',
'/manager/order/search_client');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                client_error.hidden = true;
                client_id = data[0].id;
                inp_client_first_name.value =
data[0].first_name;
                inp_client_last_name.value =
data[0].last_name;
                inp_client_fathers_name.value =
data[0].fathers_name;
                inp_client_phone_number.value =
data[0].phone_number;
            } else {
                client_error.hidden = false;
            }
        }
    });

```

```

        client_error.innerText = 'Клієнт не
знайдений';

        setTimeout(() => {
            client_error.hidden = true;
        }, 3000);
    }
};

xhr.send(post_str({
    'search': inp_client_search.value
}));
});

/**
 * Функція для додавання клієнта
 *
 * @returns {void}
 */
btn_add_client.addEventListener('click', () => {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/add_client');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            client_modal.hide();
            let data = JSON.parse(xhr.response);
            inp_client.value = sum_strs(' ',
data[0].first_name, data[0].last_name, data[0].fathers_name);
            client_id = data[0].id;
            car_div.hidden = false;
        } else {
            client_error.hidden = false;
            client_error.innerText = 'Клієнт не
доданий';

```



```

        setTimeout(() => {
            client_error.hidden = true;
        }, 3000);
    }
};

xhr.send(post_str({
    'first_name': inp_client_first_name.value,
    'last_name': inp_client_last_name.value,
    'fathers_name':
inp_client_fathers_name.value,
    'phone_number':
inp_client_phone_number.value
}));

});

// TODO: Додати перевірку на введення даних

/**
 * Функція для очистки полів модального вікна
 *
 * @returns {void}
 */
btn_car.addEventListener('click', () => {
    inp_car_search.value = '';
    inp_car_make.value = '';
    inp_car_model.value = '';
    inp_car_year.value = '';
    inp_car_license_plate.value = '';
    inp_car_description.value = '';
});

/**
 * Функція для пошуку автомобіля
 *
 * @returns {void}
 */
btn_car_search.addEventListener('click', () => {

```

```

        console.log(111);
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/manager/order/search_car');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                car_error.hidden = true;
                inp_car_make.value = data[0].make;
                inp_car_model.value = data[0].model;
                inp_car_year.value = data[0].car_year;
                inp_car_license_plate.value =
data[0].license_plate;
                inp_car_description.value =
data[0].description;
            } else {
                car_error.hidden = false;
                car_error.innerText = 'Автомобіль не
знайдений';

                setTimeout(() => {
                    car_error.hidden = true;
                }, 3000);
            }
        };
        xhr.send(post_str({
            'search': inp_car_search.value
        }));
    });

/**
 * Функція для додавання автомобіля
 *
 * @returns {void}

```

```

        */
        btn_add_car.addEventListener('click', () => {
            let xhr = new XMLHttpRequest();
            xhr.open('POST', '/manager/order/add_car');
            xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
            xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
            xhr.onload = () => {
                if (xhr.status == 200) {
                    car_modal.hide();
                    let data = JSON.parse(xhr.response);
                    inp_car.value = sum_strs(' ',
data[0].make, data[0].model, data[0].car_year);
                    car_id = data[0].id;
                    sto_list_div.hidden = false;
                    inp_sto_list.disabled = false;
                    create_order();
                } else {
                    car_error.hidden = false;
                    car_error.innerText = 'Автомобіль не
доданий';

                    setTimeout(() => {
                        car_error.hidden = true;
                    }, 3000);
                }
            };
            xhr.send(post_str({
                'make': inp_car_make.value,
                'model': inp_car_model.value,
                'car_year': inp_car_year.value,
                'license_plate':
inp_car_license_plate.value,
                'description': inp_car_description.value
            }));
        });
    });

```

```

/**
 * Функція для вибору СТО
 *
 * @returns {void}
 */
inp_sto_list.addEventListener('change', () => {
    table_div.hidden = false;
    new_task_div.hidden = false;
    btn_new_task.disabled = false;
    get_task_list();
    get_employee_list();
});

/**
 * Функція для додавання завдання
 *
 * @returns {void}
 */
btn_create.addEventListener('click', () => {
    let data = Object();
    data['name'] = inp_employee.value;
    data['task'] = inp_task.value;

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/add_task/' +
order_id);

    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;

```

```

        } else {
            error.hidden = true;
            task_modal.hide();
            get_task();
            document.getElementById('save').hidden =
false;

        }
    };

    xhr.send(post_str(data));
});

/**
 * Функція для видалення завдання
 *
 * @returns {void}
 */
btn_new_task.addEventListener('click', () => {
    inp_employee.value = '';
    inp_task.value = '';
});

/**
 * Функція для збереження замовлення
 *
 * @returns {void}
 */
btn_save.addEventListener('click', () => {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/add');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.send();
    window.location.href = '/manager/order/index'

```

```

    });

    /**
     * Функція для отримання всіх працівників СТО
     *
     * @returns {void}
     */
    function get_employee_list() {
        let xhr = new XMLHttpRequest();
        xhr.open('POST',
'/manager/order/get_employee_list');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                employee_list.innerHTML = '';
                data.forEach((item) => {
                    let option =
document.createElement('option');
                    option.value = sum_strs(' ',
item.first_name, item.last_name, item.fathers_name);
                    employee_list.appendChild(option);
                });
            }
        };
        xhr.send(post_str({
            'sto_name': inp_sto_list.value
        }));
    }

    /**
     * Функція для отримання всіх послуг
     *

```

```

        * @returns {void}
        */
function get_task_list() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST',
'/manager/order/get_task_list');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            let data = JSON.parse(xhr.response);
            task_list.innerHTML = '';
            data.forEach((item) => {
                let option =
document.createElement('option');
                option.value = item['service_name'];
                task_list.appendChild(option);
            });
        }
    };
    xhr.send();
}

/**
 * Функція для створення замовлення
 *
 * @returns {void}
 */
function create_order() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/create_order');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');

```

```

        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                order_id = data[0].id;
                document.getElementById('back').setAttri
bute('href', '/manager/order/delete/' + order_id)
            }
        };
        xhr.send(post_str({
            'client_id': client_id,
            'car_id': car_id,
        }));
    }

/**
 * Функція для отримання всіх завдань
 *
 * @returns {void}
 */
function get_task() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/get_task/' +
order_id);

    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status != 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;
        } else {
            error.hidden = true;

```



```

        let price = 0;
        let data = JSON.parse(xhr.response);
        document.getElementById('table-
body').innerHTML = '';
        for (let i = 0; i < data.length; i++) {
            price += parseInt(data[i].price);
            const tr =
document.createElement('tr');
            tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>';
            tr.innerHTML += '<td>' +
data[i].service_name + '</td>';
            tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';
            tr.innerHTML += '<td>' +
data[i].price + '</td>';
            tr.innerHTML += '<td>' +
(data[i].is_done == 't' ? 'Так' : 'Hi') + '</td>';
            tr.innerHTML += '<td>' + '<button
type="button" class="del btn" style="padding: 0;" id="' + i +
'"></button>' + '</td>';
            document.getElementById('table-
body').appendChild(tr);
        }

        let btns_del =
document.getElementsByClassName('del');
        for (let i = 0; i < btns_del.length;
i++) {
            btns_del[i].addEventListener('click'
, () => {
                if (confirm('Ви дійсно хочете
видалити цю послугу?')) {

```

```

        let xhr = new
XMLHttpRequest();

        xhr.open('POST',
'/manager/order/delete_task/' + data[i].id);
        xhr.setRequestHeader('X-
Requested-With', 'XMLHttpRequest');
        xhr.setRequestHeader("Conten
t-type", "application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status != 200) {
                error.hidden =
false;

                error.innerHTML =
'ERROR: ' + xhr.statusText;

            } else {
                error.hidden = true;
                get_task();
            }
        };
        xhr.send();
    }
    });
}
}
};
xhr.send();
}
});

/**
 * Функція для склеювання об'єкту в URL строку
 *
 * @param {Object} data - об'єкт
 * @returns {String} - склеєна строка
 */
function post_str(data) {

```

```

        let str = '';
        for (let key in data) {
            str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
        }
        return str.slice(0, -1);
    }

```

```

/**
 * Функція для склеювання строк
 *
 * @param {String} split - розділювач
 * @param {...String} theArgs - строки
 * @returns {String} - склеєна строка
 */

```

```

function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

```

</script>

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

```

</html>

Файл `app/manager/order/view_edit.php`

```
<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root
directory of this repository.
-->

<head>
    <meta charset="utf-8">
    <meta name="author" content="Dmytro Zamkovyi">
    <title><?= $title ?></title>
    <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
    <header class="d-flex flex-wrap justify-content-center py-3
mb-4 border-bottom">
```

```

        <a href="/manager/order/index" class="d-flex align-
items-center mb-3 mb-md-0 me-md-auto link-body-emphasis text-
decoration-none" style="margin-left: 16px;">
            
            <span class="fs-4">Назад</span>
        </a>

        <button id="delete" type="button" class="btn btn-
outline-danger" style="margin: 0 8px;">Видалити</button>
        <button id="save" type="button" class="btn btn-primary"
style="margin: 0 8px;">Зберегти</button>
    </header>
    <main>
        <div class="container">
            <div class="input-group mb-3">
                <span class="input-group-text">Замовник</span>
                <input type="text" class="form-control"
placeholder="Прізвище" value="<?= $order_info['first_name'] ?>
<?= $order_info['last_name'] ?> <?= $order_info['fathers_name']
?>" disabled>
            </div>

            <div class="input-group mb-3">
                <span class="input-group-text">Номер
телефону</span>
                <input type="text" class="form-control"
value="+<?= substr($order_info['phone_number'], 0, 3) ?> (<?=
substr($order_info['phone_number'], 3, 2) ?>) <?=
substr($order_info['phone_number'], 5, 3) ?>-<?=
substr($order_info['phone_number'], 8, 2) ?>-<?=
substr($order_info['phone_number'], 10, 2) ?>" disabled>
            </div>

            <div class="input-group mb-3">

```

```

        <span class="input-group-text">Дата
створення</span>
        <input type="date" class="form-control"
value="<?= $order_info['create_date'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">СТО</span>
        <input type="text" class="form-control"
value="<?= $sto_info['sto_name'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Автомобіль</span>
        <input type="text" class="form-control"
value="<?= $order_info['make'] ?> - <?= $order_info['model'] ?>
- <?= $order_info['license_plate'] ?>" disabled>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Знижка
клієнта</span>
        <?php if ($order_info['discount'] == null) : ?>
            <input type="text" class="form-control"
value="0" disabled>
        <?php else : ?>
            <input type="text" class="form-control"
value="<?= $order_info['discount'] ?>" disabled>
        <?php endif; ?>
        <label class="input-group-text"
for="discount">%</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Знижка по
замовленню</span>

```

```

        <?php if ($order_info['order_discount'] == null)
: ?>

            <input id="discount" min="0" max="100"
type="number" class="form-control" value="0">
        <?php else : ?>
            <input id="discount" min="0" max="100"
type="number" class="form-control" value="<?=
$order_info['order_discount'] ?>">
        <?php endif; ?>
        <label class="input-group-text"
for="discount">%</label>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Статус</span>
        <select id="status" class="form-control"
value="" disabled>
            <? if ($order_info['completed'] == 't') : ?>
                <option value="0"
selected>Виконано</option>
                <option value="1">Не виконано</option>
            <? else : ?>
                <option value="0">Виконано</option>
                <option value="1" selected>Не
виконано</option>
            <? endif; ?>
        </select>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-text">Оплата</span>
        <select class="form-control" disabled
autocomplete="off">
            <? if ($order_info['payment'] == 't') : ?>
                <option value="0"
selected>Оплачено</option>

```

```

        <option value="1">Не оплачено</option>
    <? else : ?>
        <option value="0">Оплачено</option>
        <option value="1" selected>Не
оплачено</option>

    <? endif; ?>
</select>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Вартість
послуг</span>
    <input id="price" type="text" class="form-
control" value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Сума
знижки</span>
    <input id="discount-price" type="text"
class="form-control" value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<div class="input-group mb-3">
    <span class="input-group-text">Вартість зі
знижкою</span>
    <input id="sum" type="text" class="form-control"
value="" disabled>
    <label class="input-group-text">⌘</label>
</div>

<p id="error" class="text-danger" hidden></p>
</div>

```



```

        <div class="container">
            <button id="new-task" type="button" class="btn btn-
primary" data-bs-toggle="modal" data-bs-target="#add-task-
model">Нове завдання</button>

            <div class="modal fade" id="add-task-model" data-bs-
backdrop="static" data-bs-keyboard="false" tabindex="-1" aria-
labelledby="staticBackdropLabel" aria-hidden="true">
                <div class="modal-dialog modal-dialog-centered
modal-lg modal-dialog-scrollable">
                    <div class="modal-content">
                        <div class="modal-header">
                            <h1 class="modal-title fs-5"
id="staticBackdropLabel">Створення акаунту клієнта</h1>
                            <button type="button" class="btn-
close" data-bs-dismiss="modal" aria-label="Close"></button>
                        </div>
                        <div class="modal-body">
                            <div class="input-group mb-3">
                                <span class="input-group-
text">Працівник</span>
                                <input id="employee" type="text"
class="form-control" list="employee-list">
                                <datalist id="employee-
list"></datalist>
                            </div>

                            <div class="input-group mb-3">
                                <span class="input-group-
text">Завдання</span>
                                <input id="task" type="text"
class="form-control" list="task-list">
                                <datalist id="task-
list"></datalist>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

```

```

        <div class="modal-footer">
            <output hidden id="error"
class="text-danger"></output>
            <button type="button" class="btn
btn-secondary" data-bs-dismiss="modal">Відміна</button>
            <button id="create" type="button"
class="btn btn-primary">Створити</button>
        </div>
    </div>
</div>
</div>
</div>
</div>

<div style="margin-left: 16px; margin-right: 16px;">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Назва послуги</th>
                <th scope="col">Виконавець</th>
                <th scope="col">Ціна</th>
                <th scope="col">Виконано</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody id="table-body"></tbody>
    </table>
</div>
</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const modal = new
bootstrap.Modal(document.getElementById('add-task-model'));
        const inp_discount =
document.getElementById('discount');

```

```

        const inp_status =
document.getElementById('status');
        const inp_price = document.getElementById('price');
        const inp_discount_price =
document.getElementById('discount-price');
        const inp_sum = document.getElementById('sum');
        const inp_employee =
document.getElementById('employee');
        const inp_task = document.getElementById('task');

        const employee_list =
document.getElementById('employee-list');
        const task_list = document.getElementById('task-
list');

        const btn_delete =
document.getElementById('delete');
        const btn_save = document.getElementById('save');
        const btn_new_task = document.getElementById('new-
task');
        const btn_create =
document.getElementById('create');

        const error = document.getElementById('error');
        const table_body = document.getElementById('table-
body');

        get_task();
        get_employee_list();
        get_task_list();

/**
 * Функція для видалення замовлення
 *
 * @returns void
 */

```

```

        btn_delete.addEventListener('click', () => {
            if (confirm('Ви дійсно хочете видалити це
замовлення?')) {
                window.location.href =
'/manager/order/delete/' + '<?= $order_id ?>';
            }
        });

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_create.addEventListener('click', () => {
    let data = Object();
    data['name'] = inp_employee.value;
    data['task'] = inp_task.value;

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/add_task/' +
'<?= $order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;
        } else {
            error.hidden = true;
            modal.hide();
            get_task();
        }
    };
});

```

```

        xhr.send(post_str(data));
    });

/**
 * Функція для збереження даних
 *
 * @returns void
 */
btn_save.addEventListener('click', () => {
    let is_error = false;
    let data = Object();
    data['discount'] = inp_discount.value;

    if (inp_discount.value == '') {
        data['discount'] = null;
    }

    if (inp_discount.value < 0 || inp_discount.value
> 100) {

        is_error = true;
        inp_discount.classList.add('is-invalid');
        return;
    }

    if (is_error) {
        error.hidden = false;
        error.innerHTML = 'Помилка вводу';
        return;
    }

    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/update/' + '<?=$order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');

```

```

        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status !== 200) {
                error.hidden = false;
                error.innerHTML = 'ERROR: ' +
xhr.statusText;

                } else {
                    error.hidden = true;
                    window.location.href =
'/manager/order/index';
                }
            };

            xhr.send(post_str(data));
        });

/**
 * Функція для очистки полів модального вікна
 *
 * @returns void
 */
btn_new_task.addEventListener('click', () => {
    inp_employee.value = '';
    inp_task.value = '';
});

/**
 * Функція для підрахунку вартості послуг
 *
 * @returns void
 */
inp_discount.addEventListener('change', () => {
    let price = parseInt(inp_price.value);
    let dis = (parseInt(inp_discount.value) / 100) +
(<?= $order_info['discount'] ?> / 100);

```

```

        if (dis > 1) {
            dis = 1;
        }
        dis = Math.round(price * dis);
        inp_discount_price.value = dis;
        inp_sum.value = price - dis;
    });

/**
 * Функція для отримання та обробки списку всіх
Завдань
 *
 * @returns void
 */
function get_task() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST', '/manager/order/get_task/<?=$order_id ?>');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status !== 200) {
            error.hidden = false;
            error.innerHTML = 'ERROR: ' +
xhr.statusText;
        } else {
            error.hidden = true;
            let price = 0;
            let data = JSON.parse(xhr.response);
            document.getElementById('table-
body').innerHTML = '';
            for (let i = 0; i < data.length; i++) {
                price += parseInt(data[i].price);
            }
        }
    }
}

```

```

        const tr =
document.createElement('tr');

        tr.innerHTML = '<th scope="row">' +
(i + 1) + '</th>';

        tr.innerHTML += '<td>' +
data[i].service_name + '</td>';

        tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';

        tr.innerHTML += '<td>' +
data[i].price + '</td>';

        tr.innerHTML += '<td>' +
(data[i].is_done == 't' ? 'Tak' : 'Hi') + '</td>';

        tr.innerHTML += '<td>' + '<button
type="button" class="del btn" style="padding: 0;" id="' + i +
'"></button>' + '</td>';

        document.getElementById('table-
body').appendChild(tr);
    }

    inp_price.value = price;
    let dis = (parseInt(inp_discount.value)
/ 100) + (<?= $order_info['discount'] ?> / 100);
    if (dis > 1) {
        dis = 1;
    }
    dis = Math.round(price * dis);
    inp_discount_price.value = dis;
    inp_sum.value = price - dis;

    let btns_del =
document.getElementsByClassName('del');
    for (let i = 0; i < btns_del.length;
i++) {

```



```

        btns_del[i].addEventListener('click'
, () => {
            if (confirm('Ви дійсно хочете
видалити цю послугу?')) {
                let xhr = new
XMLHttpRequest();
                xhr.open('POST',
'/manager/order/delete_task/' + data[i].id);
                xhr.setRequestHeader('X-
Requested-With', 'XMLHttpRequest');
                xhr.setRequestHeader("Conten
t-type", "application/x-www-form-urlencoded");
                xhr.onload = () => {
                    if (xhr.status != 200) {
                        error.hidden =
false;
                        error.innerHTML =
'ERROR: ' + xhr.statusText;
                    } else {
                        error.hidden = true;
                        get_task();
                    }
                };
                xhr.send();
            }
        });
    }
};
xhr.send();
}

/**
 * Функція для отримання та обробки списку всіх
працівників СТО
 *

```

```

        * @returns void
        */
function get_employee_list() {
    let xhr = new XMLHttpRequest();
    xhr.open('POST',
'/manager/order/get_employee_list');
    xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
    xhr.onload = () => {
        if (xhr.status == 200) {
            let data = JSON.parse(xhr.response);
            employee_list.innerHTML = '';
            data.forEach((item) => {
                let option =
document.createElement('option');
                option.value = sum_strs(' ',
item.first_name, item.last_name, item.fathers_name);
                employee_list.appendChild(option);
            });
        }
    };
    xhr.send(post_str({
        'sto_name': '<?= $sto_info['sto_name'] ?>'
    }));
}

/**
 * Функція для отримання та обробки списку всіх
послуг
 *
 * @returns void
 */
function get_task_list() {
    let xhr = new XMLHttpRequest();

```

```

        xhr.open('POST',
'/manager/order/get_task_list');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");
        xhr.onload = () => {
            if (xhr.status == 200) {
                let data = JSON.parse(xhr.response);
                task_list.innerHTML = '';
                data.forEach((item) => {
                    let option =
document.createElement('option');
                    option.value = item.service_name;
                    task_list.appendChild(option);
                });
            }
        };
        xhr.send();
    }
});

/**
 * Функція для перетворення об'єкта в URL строку
 *
 * @param {Object} data - об'єкт
 * @returns {String} - строка
 */
function post_str(data) {
    let str = '';
    for (let key in data) {
        str += encodeURIComponent(key) + '=' +
encodeURIComponent(data[key]) + '&';
    }
    return str.slice(0, -1);
}

```

```

    /**
     * Функція для склеювання строк
     *
     * @param {String} split - розділювач
     * @param {...String} theArgs - строки
     * @returns {String} - склеєна строка
     */
    function sum_strs(split, ...theArgs) {
        let total = '';
        for (const arg of theArgs) {
            if (arg == null) {
                continue;
            }
            total += arg + split;
        }
        return total.slice(0, -split.length);
    }
</script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>

```

Файл `app/manager/order/view_index.php`

```

<!DOCTYPE html>
<html data-bs-theme="dark">

<!--
License

```

Copyright (c) 2023 Dmytro Zamkovyi

You are prohibited from:

- Using any part of this code for any purpose.
- Copying or reproducing any part of this code in any form.
- Modifying or altering any part of this code.

See the full text of the license in the LICENSE file in the root directory of this repository.

-->

```
<head>
  <meta charset="utf-8">
  <meta name="author" content="Dmytro Zamkovyi">
  <title><?= $title ?></title>
  <link rel="shortcut icon" href="/data/favicon.ico"
type="image/png">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boot
strap.min.css" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN
" crossorigin="anonymous">
</head>

<body>
  <?= $head ?>
  <main style="margin-top: 16px;">
    <div class="container as castrulla">
      <button id="new-order" type="button" class="btn btn-
primary">Нове замовлення</button>
      <output <?= $success ?> id="success" class="text-
success">Замовлення успішно додано!</output>
    </div>
```

```

<hr>

<div class="container accordion" id="according-search">
  <div class="accordion-item">
    <h2 class="accordion-header">
      <button class="accordion-button collapsed"
type="button" data-bs-toggle="collapse" data-bs-
target="#collapse-search" aria-expanded="false" aria-
controls="collapseTwo">
        <h1>Пошук</h1>
      </button>
    </h2>
    <div id="collapse-search" class="accordion-
collapse collapse" data-bs-parent="#according-search">
      <div class="accordion-body">
        <div class="input-group mb-3">
          <span class="input-group-text">ПІВ
клієнта</span>
          <input id="search-first-name"
type="text" class="form-control" placeholder="Прізвище">
          <input id="search-last-name"
type="text" class="form-control" placeholder="Ім'я">
          <input id="search-father-name"
type="text" class="form-control" placeholder="Патронім">
        </div>

        <div class="input-group mb-3">
          <span class="input-group-text">Номер
телефону</span>
          <input id="search-phone-number"
type="text" class="form-control">
          <label class="input-group-text"
for="phone-number">+ZZZ (YY) XXX-XX-XX</label>
        </div>

        <div class="input-group mb-3">

```

```

        <span class="input-group-text">Homep
автомобіля</span>

        <input id="search-car" type="text"
class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Дата</span>

        <input id="search-start-date"
type="date" class="form-control">
        <label class="input-group-text">-
</label>

        <input id="search-end-date"
type="date" class="form-control">
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Статус</span>

        <select id="search-complete"
class="form-select" aria-label="Default select example">
            <option selected></option>
            <option
value="1">Завершено</option>
            <option value="2">B
процесі</option>
        </select>
    </div>

    <div class="input-group mb-3">
        <span class="input-group-
text">Оплата</span>

        <select id="search-pay" class="form-
select" aria-label="Default select example">
            <option selected></option>

```

```

        <option
value="1">Оплачено</option>

        <option value="2">Не
оплачено</option>

    </select>
</div>

<div class="gap-2 d-md-flex justify-
content-md-end">

    <button id="search" type="button"
class="btn btn-primary">Пошук</button>

    <button id="search-reset"
type="button" class="btn btn-outline-primary">Скинути</button>

</div>
</div>
</div>
</div>
</div>

<div style="margin-left: 16px; margin-right: 16px;">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">ПІБ клієнта</th>
                <th scope="col">Номер телефону</th>
                <th scope="col">Автомобіль</th>
                <th scope="col">Дата</th>
                <th scope="col">Статус</th>
                <th scope="col">Оплата</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody id="table-body"></tbody>
    </table>
</div>

```



```

</main>

<script>
    window.addEventListener('DOMContentLoaded', () => {
        const btn_create = document.getElementById('new-
order');

        const success = document.getElementById('success');

        const inp_search_first_name =
document.getElementById('search-first-name');
        const inp_search_last_name =
document.getElementById('search-last-name');
        const inp_search_father_name =
document.getElementById('search-father-name');
        const inp_search_phone_number =
document.getElementById('search-phone-number');
        const inp_search_car =
document.getElementById('search-car');
        const inp_search_start_date =
document.getElementById('search-start-date');
        const inp_search_end_date =
document.getElementById('search-end-date');
        const inp_search_complete =
document.getElementById('search-complete');
        const inp_search_pay =
document.getElementById('search-pay');
        const btn_search =
document.getElementById('search');
        const btn_search_reset =
document.getElementById('search-reset');

        const table_body = document.getElementById('table-
body');

        send();
    });

```

```

        setTimeout(() => {
            success.hidden = true;
        }, 5000);

/**
 * Функція для перенаправлення на сторінку створення
замовлення
 *
 * @returns void
 */
btn_create.addEventListener('click', () => {
    window.location.href = '/manager/order/create';
});

/**
 * Функція для пошуку замовлень
 *
 * @returns void
 */
btn_search.addEventListener('click', () => {
    let search = {};
    let is_error = false;

    if (inp_search_first_name.value !== '') {
        if (inp_search_first_name.value.length < 3
|| inp_search_first_name.value.length > 40) {
            inp_search_first_name.classList.add('is-
invalid');

            is_error = true;
        } else {
            inp_search_first_name.classList.remove('
is-invalid');

            search.first_name =
encodeURIComponent(inp_search_first_name.value);
        }
    } else {

```

```

        inp_search_first_name.classList.remove('is-
invalid');
    }

    if (inp_search_last_name.value != '') {
        if (inp_search_last_name.value.length < 3 ||
inp_search_last_name.value.length > 40) {
            inp_search_last_name.classList.add('is-
invalid');

            is_error = true;
        } else {
            inp_search_last_name.classList.remove('i
s-invalid');

            search.last_name =
encodeURIComponent(inp_search_last_name.value);
        }
    } else {
        inp_search_last_name.classList.remove('is-
invalid');
    }

    if (inp_search_father_name.value != '') {
        if (inp_search_father_name.value.length < 3
|| inp_search_father_name.value.length > 40) {
            inp_search_father_name.classList.add('is
-invalid');

            is_error = true;
        } else {
            inp_search_father_name.classList.remove(
'is-invalid');

            search.father_name =
encodeURIComponent(inp_search_father_name.value);
        }
    } else {
        inp_search_father_name.classList.remove('is-
invalid');

```

```

    }

    if (inp_search_phone_number.value != '') {
        let re = /^[ ]*\+?([0-9]{2})[-\([ ]*([0-9]{1})[-\([ ]*([0-9]{2})[-\([ ]*([0-9]{3})[- ]*([0-9]{2})[- ]*([0-9]{2})[ ]*$/gm;

        let res =
re.exec(inp_search_phone_number.value);
        if (res) {
            search.phone_number =
encodeURIComponent(res.slice(1).join(''));
            inp_search_phone_number.classList.remove
('is-invalid');
        } else {
            inp_search_phone_number.classList.add('i
s-invalid');

            inp_search_phone_number.placeholder =
'Невірний формат номеру телефону'
            is_error = true;
        }
    } else {
        inp_search_phone_number.classList.remove('is
-invalid');
    }

    if (inp_search_car.value != '') {
        search.car =
encodeURIComponent(inp_search_car.value);
    }

    if (inp_search_start_date.value != '') {
        search.start_date =
encodeURIComponent(inp_search_start_date.value);
        inp_search_start_date.classList.remove('is-
invalid');
    }

```

```

        if (inp_search_end_date.value != '') {
            inp_search_end_date.classList.remove('is-
invalid');

            search.end_date =
encodeURIComponent(inp_search_end_date.value);
        }

        if (inp_search_start_date.value != '' &&
inp_search_end_date.value != '') {
            if (inp_search_start_date.value >
inp_search_end_date.value) {
                inp_search_start_date.classList.add('is-
invalid');

                inp_search_end_date.classList.add('is-
invalid');

                is_error = true;
            } else {
                inp_search_start_date.classList.remove('
is-invalid');

                inp_search_end_date.classList.remove('is
-invalid');
            }
        } else {
            inp_search_start_date.classList.remove('is-
invalid');

            inp_search_end_date.classList.remove('is-
invalid');
        }

        if (inp_search_complete.value != '') {
            switch (inp_search_complete.value) {
                case '1':
                    search.complete =
encodeURIComponent('true');

                    break;

```

```

        case '2':
            search.complete =
encodeURIComponent('false');
            break;
        }
    }

    if (inp_search_pay.value != '') {
        switch (inp_search_pay.value) {
            case '1':
                search.pay =
encodeURIComponent('true');
                break;
            case '2':
                search.pay =
encodeURIComponent('false');
                break;
        }
    }

    send(search);
});

/**
 * Функція для скидання параметрів пошуку
 *
 * @returns void
 */
btn_search_reset.addEventListener('click', () => {
    inp_search_first_name.value = '';
    inp_search_last_name.value = '';
    inp_search_father_name.value = '';
    inp_search_phone_number.value = '';
    inp_search_car.value = '';
    inp_search_start_date.value = '';
    inp_search_end_date.value = '';

```

```

        inp_search_complete.value = '';
        inp_search_pay.value = '';
    });

    /**
     * Функція для відправки запиту на сервер
     *
     * @param {Object} search - об'єкт з параметрами
пошуку
     * @returns void
     */
    function send(search = {}) {
        let xhr = new XMLHttpRequest();
        xhr.open('POST', '/manager/order/search');
        xhr.setRequestHeader('X-Requested-With',
'XMLHttpRequest');
        xhr.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
        xhr.onload = () => {
            if (xhr.status == 200) {
                table_body.innerHTML = '';
                let data = JSON.parse(xhr.responseText);
                for (let i = 0; i < data.length; i++) {
                    let tr =
document.createElement('tr');
                    tr.classList.add('col-12', 'col-lg-
auto', 'my-2', 'justify-content-center', 'my-md-0');
                    tr.innerHTML = '<th score="row">' +
(i + 1) + '</th>';
                    tr.innerHTML += '<td>' + sum_strs('
', data[i].first_name, data[i].last_name, data[i].fathers_name)
+ '</td>';
                    tr.innerHTML += '<td>' + '+' +
data[i].phone_number.slice(0, 3) + ' (' +
data[i].phone_number.slice(3, 5) + ') ' +
data[i].phone_number.slice(5, 8) + '-' +

```

```

data[i].phone_number.slice(8, 10) + '-' +
data[i].phone_number.slice(10, 12) + '</td>';
        tr.innerHTML += '<td>' + sum_strs('
- ', data[i].make, data[i].model, data[i].license_plate) +
'</td>';

        tr.innerHTML += '<td>' +
data[i].create_date + '</td>';
        tr.innerHTML += '<td>' +
(data[i].completed ? 'Завершено' : 'В процесі') + '</td>';
        tr.innerHTML += '<td>' +
(data[i].payment ? 'Оплачено' : 'Не оплачено') + '</td>';
        tr.innerHTML += '<td>' + '<button
type="button" class="edit btn" style="padding: 0;"
onclick=edit(' + data[i]['id'] + ')></button>'
+ '</td>'

        table_body.appendChild(tr);
    }
}
};
xhr.send(post_str(search));
}
});

/**
 * Функція для перетворення об'єкта в строку
 *
 * @param {Object} obj - об'єкт
 * @returns {String} - строка
 */
function post_str(obj) {
    let str = '';
    for (let p in obj) {
        str += encodeURIComponent(p) + "=" +
encodeURIComponent(obj[p]);

```



```

        str += '&';
    }
    return str.slice(0, -1);
}

/**
 * Функція для склеювання строк
 *
 * @param {String} split - розділювач
 * @param {...String} theArgs - строки
 * @returns {String} - склеєна строка
 */
function sum_strs(split, ...theArgs) {
    let total = '';
    for (const arg of theArgs) {
        if (arg == null) {
            continue;
        }
        total += arg + split;
    }
    return total.slice(0, -split.length);
}

/**
 * Функція для перенаправлення на сторінку редагування
працівника
 *
 * @param {Number} i - id працівника
 * @returns void
 */
function edit(i) {
    window.location.href = '/manager/order/edit/' + i;
}
</script>

```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootst
rap.bundle.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL
" crossorigin="anonymous"></script>
</body>

</html>
```