

1. ВИЗНАЧИТИ ОСНОВНІ ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Увага! У даних вказівках буде продемонстровано створення лише спрощеного фрагменту застосунку для роботи з базою даних.

Основні функціональні можливості фрагменту web-застосунку, який призначений для роботи з базою даних «supply», наведені у вигляді UML-діаграми прецедентів (рисунок 1).

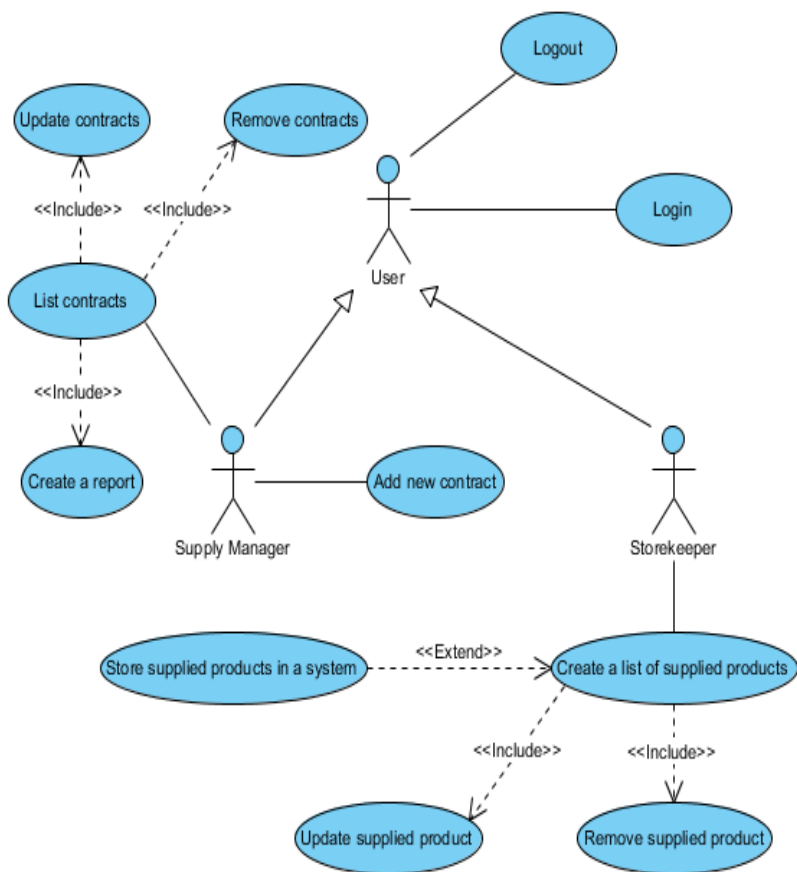


Рисунок 1

2. РОЗРОБИТИ СТОРІНКУ ДЛЯ АВТОРИЗАЦІЇ КОРИСТУВАЧІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Усі сторінки web-застосунку необхідно розмістити у каталозі «xampp/htdocs/supply».

Перш ніж приступати до створення сторінки авторизації, необхідно розробити функціональність програмного забезпечення щодо встановлення з'єднання з базою даних. Для цього створити файл «connect.php» з наступним вмістом.

```
<?php
function db_conn() {
    $server = "localhost";
    $user = $_SESSION["user"];
    $pass = $_SESSION["pass"];
    $db = "supply";

    $conn = @mysqli_connect($server, $user, $pass, $db);

    if (!$conn) {
        session_unset();
        session_destroy();

        die("Connection failed: " . mysqli_connect_error());
    }

    return $conn;
}
?>
```

Крім того, необхідно розробити основну сторінку web-застосунку, для чого створити файл «index.php» з наступним вмістом.

```
1  <?php
2  session_start();
3
4  require_once("connect.php");
5
6  $conn = NULL;
7
8  # check for a user session
9  if (isset($_SESSION["user"])) {
10     $conn = db_conn();
11     include("action.php");
12 } else {
13     # redirected to login page if the user is not set
14     header("location: login.php");
15 }
16 ?>
```

Рядки 2 – 4 містять початок користувацького сеансу та підключення файлу, що містить функцію «db_conn()» для встановлення з'єднання з базою даних. Рядки 9 – 15 містять перевірку наявності користувацького сеансу та підключення до бази даних. В разі, якщо користувач не був авторизований, він буде перенаправлений на сторінку авторизації (рядок 14). Рядок 11 визначає підключення файлу, який містить обробку форм додавання, оновлення та видалення даних, він буде створений пізніше.

```

17 <!DOCTYPE html>
18 <html>
19 <head>
20 <title>Supply</title>
21 </head>
22 <body>
23 <p>
24 <b>User:</b> <i>= $_SESSION["user"] ?&gt;&lt;/i&gt; | &lt;a href="logout.php"&gt;Logout&lt;/a&gt;
25 &lt;/p&gt;
26 &lt;?php
27 # display content depending on the user type
28 if ($_SESSION["user"] == "manager") {
29     include("manager.php");
30 }
31
32 if ($_SESSION["user"] == "storekeeper") {
33     include("storekeeper.php");
34 }
35 ?&gt;
36 &lt;/body&gt;
37 &lt;/html&gt;
38 &lt;?php
39 mysql_close($conn);
</pre

```

Наступні рядки (17 – 39) визначають вигляд головної сторінки: інформацію про поточного користувача (рисунок 2), вміст сторінки відповідно до типу користувача, відключення з'єднання з базою даних (рядок 39). У рядку 24 визначене посилання, яке дозволяє видалити усі змінні сеансу та завершити сеанс використання застосунку користувачем. Для цього використовується файл «logout.php».

User: *manager* | [Logout](#)

Рисунок 2

```

<?php
session_start();

# remove session variables and destroy a session
session_unset();
session_destroy();

header("location: login.php");

```

Сторінка авторизації користувача зберігається у файлі «login.php».

```
1  <?php
2  session_start();
3
4  # process login form
5  if (isset($_POST["login"])) {
6      session_unset();
7
8      # set user session variables
9      $_SESSION["user"] = $_POST["user"];
10     $_SESSION["pass"] = $_POST["pass"];
11
12     header("location: index.php");
13 } else {
14     # redirect to a home page if user is already signed in
15     if (isset($_SESSION["user"])) {
16         header("location: index.php");
17     }
18 }
19 ?>
```

Рядки 9 та 10 визначають запис змінних сеансу, які містять дані про обліковий запис користувача. Саме ці змінні використовуються у файлі «connect.php» для встановлення з'єднання з базою даних за допомогою функції «mysqli_connect()». Якщо сеанс користувача вже був встановлений, він буде перенаправлений на головну сторінку «index.php» (рядки 15 – 17).

```
20 <!DOCTYPE html>
21 <html>
22 <head>
23     <title>Login</title>
24 </head>
25 <body>
26     <h3>Supply Application Login</h3>
27     <form method="post" action="login.php">
28         <p>
29             <b>User name</b>
30         </p>
31         <p>
32             <input type="text" name="user" required />
33         </p>
34         <p>
35             <b>Password</b>
36         </p>
37         <p>
38             <input type="password" name="pass" required />
39         </p>
40         <p>
41             <input type="submit" name="login" value="Login" />
42         </p>
43     </form>
44 </body>
45 </html>
```

У рядках 20 – 45 визначена статична структура сторінки авторизації користувача, яка містить відповідну форму з необхідними елементами користувацького інтерфейсу (рисунок 3).

Supply Application Login

User name

Password

Login

Рисунок 3

Програмний код файлу «connect.php»:

```
<?php
function db_conn() {
    $server = "localhost";
    $user = $_SESSION["user"];
    $pass = $_SESSION["pass"];
    $db = "supply";

    $conn = @mysqli_connect($server, $user, $pass, $db);

    if (!$conn) {
        session_unset();
        session_destroy();

        die("Connection failed: " . mysqli_connect_error());
    }

    return $conn;
}
?>
```

Программный код файла «login.php»:

```
<?php
session_start();

# process login form
if (isset($_POST["login"])) {
    session_unset();

    # set user session variables
    $_SESSION["user"] = $_POST["user"];
    $_SESSION["pass"] = $_POST["pass"];

    header("location: index.php");
} else {
    # redirect to a home page if user is already signed in
    if (isset($_SESSION["user"])) {
        header("location: index.php");
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h3>Supply Application Login</h3>
    <form method="post" action="login.php">
        <p>
            <b>User name</b>
        </p>
        <p>
            <input type="text" name="user" required />
        </p>
        <p>
            <b>Password</b>
        </p>
        <p>
            <input type="password" name="pass" required />
        </p>
        <p>
            <input type="submit" name="login" value="Login" />
        </p>
```

```
</form>
</body>
</html>
```

Програмный код файла «logout.php»:

```
<?php
session_start();

# remove session variables and destroy a session
session_unset();
session_destroy();

header("location: login.php");
?>
```

Програмный код файла «index.php»:

```
<?php
session_start();

require_once("connect.php");

$conn = NULL;

# check for a user session
if (isset($_SESSION["user"])) {
    $conn = db_conn();
    include("action.php");
} else {
    # redired to login page if the user is not set
    header("location: login.php");
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Supply</title>
</head>
<body>
    <p>
        <b>User:</b> <i>= $_SESSION["user"] ?&gt;&lt;/i&gt; | &lt;a
href="logout.php"&gt;Logout&lt;/a&gt;</pre
```

```
</p>
<?php
# display content depending on the user type
if ($_SESSION["user"] == "manager") {
    include("manager.php");
}

if ($_SESSION["user"] == "storekeeper") {
    include("storekeeper.php");
}
?>
</body>
</html>
<?php
mysqli_close($conn);
?>
```


3. РОЗРОБИТИ ФУНКЦІОНАЛЬНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТИ МЕНЕДЖЕРА З ЗАКУПІВЕЛЬ

Сторінка, що містить функціональність програмного забезпечення для роботи менеджера з закупівель, міститься у файлі «manager.php».

```
1 <?php
2 # check for a user session
3 if (!isset($_SESSION["user"])) {
4     header("Location: login.php");
5 }
6 }
7
8 <h3>Contracts</h3>
9
10 <p>
11 <?php
12 # if the page is in record's create/update or delete mode (action parameter is set) - show 'back' link
13 if (isset($_GET["action"]) && ($_GET["action"] == "create" || $_GET["action"] == "update"
14 || $_GET["action"] == "delete")) {
15     ?>
16     <a href="index.php">Back</a>
17 <?php
18 # otherwise - show 'new record' link
19 } else {
20     ?>
21     <a href="index.php?action=create">New contract</a>
22 <?php
23 }
24 </p>
```

Рядки 1 – 24 містять перевірку наявності користувацького сеансу, а також режиму роботи з даними про договори (створення, оновлення або видалення), від чого залежить елемент інтерфейсу – посилання «New contract», призначене для створення нового договору (рисунок 4), або «Back» – для повернення до перегляду даних про усі договори (рисунок 5).

Contracts

[New contract](#)

Contract number	Contract date	Supplier	Note	Action
1	2018-09-01 00:00:00	Petrov Pavlo Petrovych	Order 34 on 30.08.2018	Update Delete
2	2018-09-10 00:00:00	Petrov Pavlo Petrovych	Invoice 08-78 on 28.08.2018	Update Delete
3	2018-09-23 00:00:00	Ivanov Illia Illych	Order 56 on 28.08.2018	Update Delete
4	2018-09-24 00:00:00	Interfruit Ltd.	Order 74 on 11.09.2018	Update Delete
5	2018-10-02 00:00:00	Interfruit Ltd.	Invoice 09-12 on 21.09.2018	Update Delete
7	2018-12-27 13:30:04	Petrov Pavlo Petrovych		Update Delete
13	2019-01-10 13:20:48	Transservice LLC	Order #9876	Update Delete

Рисунок 4

[Back](#)

Supplier

Petrov Pavlo Petrovych ▼

Note

Save

Рисунок 5

Рядки 26 – 99 містять перевірку режимів створення нового запису (рисунок 5), оновлення (рисунок 6) або видалення існуючого запису (рисунок 7) та відображення відповідних форм з певними елементами інтерфейсу користувача.

```
26 <?php
27 # check for action parameter
28 # show create/update or delete form if it is set
29 if (isset($_GET["action"]) && ($_GET["action"] == "create" || $_GET["action"] == "update"
30 || $_GET["action"] == "delete")) {
31 ?>
32
33 <form method="post" action="index.php">
34 <input type="hidden" value="<?=$_GET["id"] ?>" name="contract_number" />
35 <?php
36 # if the current mode is create/update
37 # show corresponding form with the required fields and buttons
38 if ($_GET["action"] == "create" || $_GET["action"] == "update") {
39 ?>
40 <p>
41 <b>Supplier</b>
42 </p>
43 <p>
44 <select name="supplier_id">
45 <?php
46 # retrieve suppliers ids/info to display select control
47 $sql = "SELECT * FROM supplier_info";
48 $result = mysqli_query($conn, $sql);
49
50 while ($row = mysqli_fetch_assoc($result)) {
51 ?><option value="<?=$_GET["supplier_id"] ?>"><?=$_GET["Info"] ?></option><?php
52 }
53 ?>
54 </select>
55 </p>
```

```

55 <p>
56 <b>Note</b>
57 </p>
58 <p>
59 <?php
60 # retrieve and display contract note of the updated contract
61 if (isset($_GET["action"]) && $_GET["action"] == "update") {
62     $contract_number = $_GET["id"];
63
64     $sql = "SELECT contract_note FROM contract WHERE contract_number = {$contract_number}";
65     $result = mysqli_query($conn, $sql);
66     $row = mysqli_fetch_assoc($result);
67 }
68 <?
69 <textarea name="contract_note" rows="5" cols="50">= $row["contract_note"] ?&gt;&lt;/textarea&gt;
70 &lt;/p&gt;
71 &lt;p&gt;
72 &lt;?php
73 # set proper names for create/update buttons
74 if (isset($_GET["action"]) &amp;&amp; $_GET["action"] == "create") {
75     &lt;?
76     &lt;input type="submit" name="create_contract" value="Save" /&gt;
77 &lt;?php
78 } else if (isset($_GET["action"]) &amp;&amp; $_GET["action"] == "update") {
79     &lt;?
80     &lt;input type="submit" name="update_contract" value="Save" /&gt;
81 &lt;?php
82 }
83 &lt;?
84 &lt;/p&gt;
</pre

```

85 <?php
86 # if the current mode is delete
87 # display the corresponding question and button
88 } else if ($_GET["action"] == "delete") {
89 <?
90 Delete the contract #<?= $_GET["id"] ?>
91 <p>
92 <input type="submit" name="delete_contract" value="Continue" />
93 </p>
94 <?php
95 }
96 <?
97 </form>
98 <?php
99 } else {

```


```

Supplier

Transservice LLC ▼

Note

Order #9876

Save

Рисунок 6

[Back](#)

Delete the contract #13?

Continue

Рисунок 7

Рядки 100 – 133, в свою чергу, визначають таблицю з даними про договори та відповідними посиланнями (стовпчик «Action»), призначеними для маніпулювання цими даними (рисунок 4).

Рядки 135 – 179 містять визначення додаткової таблиці, призначеної для відображення переліку поставлених товарів за певним договором (рисунок 8). Для демонстрації даної таблиці виконується необхідна перевірка режиму перегляду даних про договори (рядки 137 – 138).

```
100 | ->
101 |
102 |     <table border="1">
103 |         <tr>
104 |             <th>Contract number</th>
105 |             <th>Contract date</th>
106 |             <th>Supplier</th>
107 |             <th>Note</th>
108 |             <th>Action</th>
109 |         </tr>
110 |
111 |         <?php
112 |             # retrieve and display data about contracts
113 |             $sql = "SELECT contract_supplier.*,
114 |                 (SELECT contract_note FROM contract WHERE contract_number = contract_supplier.contract_number) AS 'note'
115 |                 FROM contract_supplier";
116 |             $result = mysqli_query($conn, $sql);
117 |
118 |             while ($row = mysqli_fetch_assoc($result)) {
119 |                 <tr>
120 |                     <td><a href="index.php?action=info&id=<?> $row["contract_number"] <?>"><?> $row["contract_number"] <?></a></td>
121 |                     <td><?> $row["contract_date"] <?></td>
122 |                     <td><?> $row["Supplier"] <?></td>
123 |                     <td><?> $row["note"] <?></td>
124 |                     <td>
125 |                         <a href="index.php?action=update&id=<?> $row["contract_number"] <?>">Update</a>
126 |                         <a href="index.php?action=delete&id=<?> $row["contract_number"] <?>">Delete</a>
127 |                     </td>
128 |                 </tr>
129 |             }
130 |         <?php
131 |     </table>
132 |
133 | }
```

```

135 # if the action mode is info
136 # display data about supplied products for a selected contract
137 if (isset($_GET["action"]) && $_GET["action"] == "info") {
138     $contract_number = $_GET["id"];
139 }
140 <h3>Supplied products by contract #<?=$_contract_number ?></h3>
141 <p>
142     <a href="index.php">Hide</a>
143 </p>
144 <?php
145     # retrieve data about selected products
146     $sql = "SELECT supplied_product, supplied_amount, supplied_cost
147           FROM supplied
148           WHERE contract_number = {$_contract_number}";
149     $result = mysqli_query($conn, $sql);
150
151     # check the size of a result set
152     if (mysqli_num_rows($result) > 0) {
153     }
154     <table border="1">
155     <tr>
156         <th>Product</th>
157         <th>Amount</th>
158         <th>Cost</th>
159     </tr>
160 <?php
161
162     # display products if the contract is not empty
163     while ($row = mysqli_fetch_assoc($result)) {
164     }
165     <tr>
166         <td><?=$_row["supplied_product"] ?></td>
167         <td><?=$_row["supplied_amount"] ?></td>
168         <td><?=$_row["supplied_cost"] ?></td>
169     </tr>
170     <?php
171 } else {
172     # if the result set is empty print the following message
173     echo "Contract is empty";
174 }
175 }
176 </table>
177 <?php
178 }
179 }

```

Contract number	Contract date	Supplier	Note	Action
1	2018-09-01 00:00:00	Petrov Pavlo Petrovych	Order 34 on 30.08.2018	Update Delete
2	2018-09-10 00:00:00	Petrov Pavlo Petrovych	Invoice 08-78 on 28.08.2018	Update Delete
3	2018-09-23 00:00:00	Ivanov Illia Illych	Order 56 on 28.08.2018	Update Delete
4	2018-09-24 00:00:00	Interfruit Ltd.	Order 74 on 11.09.2018	Update Delete
5	2018-10-02 00:00:00	Interfruit Ltd.	Invoice 09-12 on 21.09.2018	Update Delete
7	2018-12-27 13:30:04	Petrov Pavlo Petrovych		Update Delete
13	2019-01-10 13:20:48	Transservice LLC	Order #9876	Update Delete

Supplied products by contract #4

[Hide](#)

Product	Amount	Cost
Audio Player	22	320.00
Printer	41	332.50
TV	56	990.00

Рисунок 8

Программний код файлу «manager.php»:

```

<?php
# check for a user session
if (!isset($_SESSION["user"])) {
    header\("location: login.php"\);
}
?>

<h3>Contracts</h3>
<p>
<?php
# if the page is in record's create/update or delete mode (action parameter
is set) - show 'back' link
if (isset($_GET["action"]) && ($_GET["action"] == "create" || $_GET["ac
tion"] == "update"
|| $_GET["action"] == "delete")) {
?>
    <a href="index.php">Back</a>
<?php
# otherwise - show 'new record' link
} else {
?>
    <a href="index.php?action=create">New contract</a>
    <a href="index.php?action=export">Export data</a>
<?php
}
?>

```

</p>

<?php

check for action parameter

show create/update or delete form if it is set

if (isset(\$_GET["action"]) && (\$_GET["action"] == "create" || \$_GET["action"] == "update"

|| \$_GET["action"] == "delete")) {

?>

<form method="post" action="index.php">

<input type="hidden" value="<?= \$_GET["id"] ?>"

name="contract_number" />

<?php

if the current mode is create/update

show corresponding form with the required fields and buttons

if (\$_GET["action"] == "create" || \$_GET["action"] == "update") {

?>

<p>

Supplier

</p>

<p>

<select name="supplier_id">

<?php

retrieve suppliers ids/info to display select control

\$sql = "SELECT * FROM supplier_info";

\$result = mysqli_query(\$conn, \$sql);

while (\$row = mysqli_fetch_assoc(\$result)) {

?><option

value="<?= \$row["supplier_id"] ?>"><?= \$row["Info"] ?></option><?php

}

?>

</select>

</p>

<p>

Note

</p>

<p>

<?php

retrieve and display contract note of the updated contract

if (isset(\$_GET["action"]) && \$_GET["action"] == "update") {

\$contract_number = \$_GET["id"];

```

    $sql = "SELECT contract_note FROM contract WHERE
contract_number = {$contract_number}";
    $result = mysqli_query($conn, $sql);
    $row = mysqli_fetch_assoc($result);
}
?>
<textarea name="contract_note" rows="5"
cols="50"><?= $row["contract_note"] ?></textarea>
</p>
<p>
<?php
# set proper names for create/update buttons
if (isset($_GET["action"])) && $_GET["action"] == "create") {
?>
    <input type="submit" name="create_contract" value="Save" />
<?php
} else if (isset($_GET["action"])) && $_GET["action"] == "update") {
?>
    <input type="submit" name="update_contract" value="Save" />
<?php
}
?>
</p>
<?php
# if the current mode is delete
# display the corresponding question and button
} else if ($_GET["action"] == "delete") {
?>
    <b>Delete the contract #<?= $_GET["id"] ?>?</b>
    <p>
        <input type="submit" name="delete_contract" value="Continue" />
    </p>
<?php
}
?>
</form>
<?php
} else {
?>
<table border="1">
<tr>
<th>Contract number</th>
<th>Contract date</th>
<th>Supplier</th>

```



```

        <th>Note</th>
        <th>Action</th>
    </tr>
<?php
    # retrieve and display data about contracts
    $sql = "SELECT contract_supplier.*,
        (SELECT contract_note FROM contract WHERE contract_number =
        contract_supplier.contract_number) AS `note`
        FROM contract_supplier";
    $result = mysqli_query($conn, $sql);

    while ($row = mysqli_fetch_assoc($result)) {
        ?>
        <tr>
            <td><a
href="index.php?action=info&id=<?= $row["contract_number"] ?>"><?= $
row["contract_number"] ?></a></td>
            <td><?= $row["contract_date"] ?></td>
            <td><?= $row["Supplier"] ?></td>
            <td><?= $row["note"] ?></td>
            <td>
                <a
href="index.php?action=update&id=<?= $row["contract_number"] ?>">Up
date</a>
                <a
href="index.php?action=delete&id=<?= $row["contract_number"] ?>">Del
ete</a>
            </td>
        </tr>
    </?php
    }
    ?>
</table>
<?php
}

# if the action mode is info
# display data about supplied products for a selected contract
if (isset($_GET["action"]) && $_GET["action"] == "info") {
    $contract_number = $_GET["id"];
    ?>
    <h3>Supplied products by contract #<?= $contract_number ?></h3>
    <p>
    <a href="index.php">Hide</a>

```

```

</p>
<?php
# retrieve data about selected products
$sql = "SELECT supplied_product, supplied_amount, supplied_cost
FROM supplied
WHERE contract_number = {$contract_number}";
$result = mysqli_query($conn, $sql);

# check the size of a result set
if (mysqli_num_rows($result) > 0) {
    ?>
    <table border="1">
        <tr>
            <th>Product</th>
            <th>Amount</th>
            <th>Cost</th>
        </tr>
    <?php
    # display products if the contract is not empty
    while ($row = mysqli_fetch_assoc($result)) {
        ?>
        <tr>
            <td><?= $row["supplied_product"] ?></td>
            <td><?= $row["supplied_amount"] ?></td>
            <td><?= $row["supplied_cost"] ?></td>
        </tr>
    <?php
    }
} else {
    # if the result set is empty print the following message
    echo "Contract is empty";
}
?>
</table>
<?php
}
?>

```

4. РОЗРОБИТИ ФУНКЦІОНАЛЬНІСТЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТИ ПРАЦІВНИКА СКЛАДУ

Сторінка, що містить функціональність програмного забезпечення для роботи працівника складу, міститься у файлі «storekeeper.php».

Рядки 1 – 14 містять перевірку наявності користувацького сеансу, а також наявності змінної сеансу – масиву, до якого будуть записані товари, поставлені на склад, але ще не збережені у базі даних.

```
1 <?php
2 # check for a user session
3 if (!isset($_SESSION["user"])) {
4     header("location: login.php");
5 }
6
7 # initialize array of delivered but not stored products
8 # such array is implemented as the session variable
9 if (!isset($_SESSION["supplied_products"])) {
10     $_SESSION["supplied_products"] = array();
11 }
12 ?>
13
14 <h3>Supplied products</h3>
```

У рядках 16 – 72 визначається таблиця поставлених на склад товарів.

```
16 <?php
17 # check for awaiting deliveries (is there any empty contracts)
18 $sql = "SELECT * FROM contract_supplier
19 WHERE contract_number NOT IN (SELECT contract_number FROM supplied)";
20 $result = mysqli_query($conn, $sql);
21
22 # if awaiting deliveries exist
23 # display a corresponding form
24 if (mysqli_num_rows($result) > 0) {
25     # check session array of delivered but not stored products
26     # if there are any products - display the form used to store supplied products
27     if (sizeof($_SESSION["supplied_products"]) > 0) {
28         ?>
29         <form method="post" action="index.php">
30             <p>
31                 <b>by contract</b>
32                 <select name="contract_number">
33                     <?php
34                         # display the combo box with awaiting orders
35                         while ($row = mysqli_fetch_assoc($result)) {
36                             ?><option value="<?= $row["contract_number"] ?>">
37                                 <?= $row["contract_number"] . " - " . $row["Supplier"] .
38                                 " (" . $row["contract_date"] . ")" ?></option><?php
39                             }
40                             ?>
41                         </select>
42                     </p>
43                     <table border="1">
44                         <tr>
45                             <th>Product</th>
46                             <th>Amount</th>
47                             <th>Cost</th>
48                             <th>Action</th>
49                         </tr>
```

При цьому, виконується перевірка наявності товарів у масиві (змінна сеансу) та вивід форми (рисунок 9), що дозволяє записати прийняті товари у базу даних (рядки 27 – 67).

```

50      <?php
51      # display the session array of delivered products
52      foreach ($_SESSION["supplied_products"] as $key => $value) {
53          <?
54          <tr>
55              <td><?=$key ?></td>
56              <td><?=$value["amount"] ?></td>
57              <td><?=$value["cost"] ?></td>
58              <td><a href="index.php?supplied=remove&product=<?=$key ?>">Remove</a></td>
59          </tr>
60      <?php
61      }
62      <?
63      </table>
64      <p>
65          <input type="submit" name="save_products" value="Store products" />
66      </p>
67      </form>
68      <?php
69      } else {
70          echo "Add supplied products";
71      }
72      <?

```

Також перевіряється наявність очікуваних поставок (якщо є так звані «пусті» договори, що були укладені, але за якими ще не були поставлені товари) у рядках 17 – 24. В разі наявності таких договорів, демонструється форма (рисунок 10) для додавання прийнятого товару (рядки 73 – 103).

```

73      <p>
74          <b>New product</b>
75      </p>
76      <form method="post" action="index.php">
77          <table border="1">
78              <tr>
79                  <th>Product</th>
80                  <th>Amount</th>
81                  <th>Cost</th>
82              </tr>
83              <tr>
84                  <td>
85                      <input type="text" name="supplied_product" required />
86                  </td>
87                  <td>
88                      <input type="number" name="supplied_amount" min="0.01" step="0.01" value="0.01" required />
89                  </td>
90                  <td>
91                      <input type="number" name="supplied_cost" min="0.01" step="0.01" value="0.01" required />
92                  </td>
93              </tr>
94          </table>
95          <p>
96              <input type="submit" name="add_product" value="Add product">
97          </p>
98      </form>
99      <?php
100      } else {
101          echo "There are no awaiting deliveries";
102      }
103      <?

```

Supplied products

by contract 13 - Transservice LLC (2019-01-10 13:20:48) ▼

| Product | Amount | Cost | Action |
|---------|--------|--------|------------------------|
| TV | 15 | 900 | Remove |
| Camera | 30 | 1200 | Remove |
| Watch | 200 | 399.99 | Remove |

Store products

Рисунок 9

New product

| Product | Amount | Cost |
|-------------------|--------|------|
| Bluetooth Speaker | 99 | 120 |

Add product

Рисунок 10

Программный код файла «storekeeper.php»:

```
<?php
# check for a user session
if (!isset($_SESSION["user"])) {
    header("location: login.php");
}

# initialize array of delivered but not stored products
# such array is implemented as the session variable
if (!isset($_SESSION["supplied_products"])) {
    $_SESSION["supplied_products"] = array();
}
?>

<h3>Supplied products</h3>
<?php
# check for awaiting deliveries (is there any empty contracts)
$sql = "SELECT * FROM contract_supplier
WHERE contract_number NOT IN (SELECT contract_number FROM
supplied)";
$result = mysqli_query($conn, $sql);
```

```

# if awaiting deliveries exist
# display a corresponding form
if (mysqli_num_rows($result) > 0) {
    # check session array of delivered but not stored products
    # if there are any products - display the form used to store supplied
    products
    if (sizeof($_SESSION["supplied_products"]) > 0) {
        ?>
        <form method="post" action="index.php">
        <p>
            <b>by contract</b>
            <select name="contract_number">
            <?php
                # display the combo box with awaiting orders
                while ($row = mysqli_fetch_assoc($result)) {
                    ?><option value="<?=$row["contract_number"] ?>">
                        <?=$row["contract_number"] . " - " . $row["Supplier"] .
                            " (" . $row["contract_date"] . ")" ?></option><?php
                }
                ?>
            </select>
        </p>
        <table border="1">
            <tr>
                <th>Product</th>
                <th>Amount</th>
                <th>Cost</th>
                <th>Action</th>
            </tr>
            <?php
                # display the session array of delivered products
                foreach ($_SESSION["supplied_products"] as $key => $value) {
                    ?>
                    <tr>
                        <td><?=$key ?></td>
                        <td><?=$value["amount"] ?></td>
                        <td><?=$value["cost"] ?></td>
                        <td><a
href="index.php?supplied=remove&product=<?=$key ?>">Remove</a></td>
                    </tr>
                <?php
                }
                ?>

```

```

        </table>
        <p>
            <input type="submit" name="save_products" value="Store products"
    />
        </p>
    </form>
<?php
} else {
    echo "Add supplied products";
}
?>
<p>
    <b>New product</b>
</p>
<form method="post" action="index.php">
    <table border="1">
        <tr>
            <th>Product</th>
            <th>Amount</th>
            <th>Cost</th>
        </tr>
        <tr>
            <td>
                <input type="text" name="supplied_product" required />
            </td>
            <td>
                <input type="number" name="supplied_amount" min="0.01"
step="0.01" value="0.01" required />
            </td>
            <td>
                <input type="number" name="supplied_cost" min="0.01"
step="0.01" value="0.01" required />
            </td>
        </tr>
    </table>
    <p>
        <input type="submit" name="add_product" value="Add product">
    </p>
</form>
<?php
} else {
    echo "There are no awaiting deliveries";
}
?>

```

5. РОЗРОБИТИ ФУНКЦІОНАЛЬНІСТЬ ДЛЯ ФОРМУВАННЯ ЗВІТУ У ФОРМАТІ EXCEL ЩОДО ОБСЯГІВ ПОСТАВЛЕНОЇ ПРОДУКЦІЇ ЗА ПЕВНИЙ ПЕРІОД

Реалізація даної функціональної можливості буде знаходитися також у файлі «action.php», який містить обробку користувацьких форм.

Рядки 1 – 34 даного файлу містять обробки форм, які призначені для створення записів про договори, а також оновлення та видалення існуючих записів. Слід звернути увагу, що для виконання операцій створення, оновлення та видалення записів з таблиці contract використовується створена раніше збережена процедура «sp_contract_ops».

У рядках 36 – 60 обробляються форми, які призначені для створення запису про поставлений, але ще не збережений у базі даних товар, а також видалення таких записів з масиву, що зберігається в якості змінної сеансу користувача.

```
1 <?php
2 # process request to create contract
3 if (isset($_POST["create_contract"])) {
4     $supplier_id = $_POST["supplier_id"];
5     $contract_note = $_POST["contract_note"];
6
7     # use the stored procedure created earlier
8     $sql = "CALL sp_contract_ops('i', 0, '', ($supplier_id), '{$contract_note}')";
9     mysqli_query($conn, $sql);
10
11     header("location: index.php");
12 }
13
14 # process request to delete contract
15 if (isset($_POST["delete_contract"])) {
16     $contract_number = $_POST["contract_number"];
17
18     $sql = "CALL sp_contract_ops('d', ($contract_number), '', 0, '')";
19     mysqli_query($conn, $sql);
20
21     header("location: index.php");
22 }
23
24 # process request to update contract
25 if (isset($_POST["update_contract"])) {
26     $contract_number = $_POST["contract_number"];
27     $supplier_id = $_POST["supplier_id"];
28     $contract_note = $_POST["contract_note"];
29
30     $sql = "CALL sp_contract_ops('u', ($contract_number), CURRENT_TIMESTAMP(), ($supplier_id), '{$contract_note}')";
31     mysqli_query($conn, $sql);
32
33     header("location: index.php");
34 }
```



```

36 # process request to insert new record into session array of delivered products
37 if (isset($_POST["add_product"])) {
38     $supplied_product = $_POST["supplied_product"];
39     $supplied_amount = $_POST["supplied_amount"];
40     $supplied_cost = $_POST["supplied_cost"];
41
42     if (!empty($supplied_product) && !empty($supplied_amount) && !empty($supplied_cost)) {
43         if (is_numeric($supplied_amount) && is_numeric($supplied_cost)) {
44             if ($supplied_amount > 0 && $supplied_cost > 0) {
45                 $_SESSION["supplied_products"][$supplied_product] = array("amount" => $supplied_amount,
46                                     "cost" => $supplied_cost);
47             }
48         }
49     }
50
51     header("location: index.php");
52 }
53
54 # process request to remove a record from the session array
55 if (isset($_GET["supplied"]) && $_GET["supplied"] == "remove") {
56     $supplied_product = $_GET["product"];
57     unset($_SESSION["supplied_products"][$supplied_product]);
58     header("location: index.php");
59 }
60 }

```

Рядки 62 – 103 демонструють збереження поставлених товарів до бази даних. Слід звернути увагу на те, що створення записів про товари, поставлені за певним договором, у таблиці «supplied», виконується всередині транзакції, оскільки часткове, в силу будь-яких обставин, перенесення даних про прийняті товари з сеансової змінної до оперативної бази даних є неприйнятним.

```

62 # process request to store delivered products into the database
63 if (isset($_POST["save_products"])) {
64     $contract_number = $_POST["contract_number"];
65
66     # begin transaction
67     mysqli_query($conn, "SET AUTOCOMMIT = 0");
68     mysqli_query($conn, "START TRANSACTION");
69
70     $failed = false;
71
72     foreach ($_SESSION["supplied_products"] as $key => $value) {
73         $amount = $value["amount"];
74         $cost = $value["cost"];
75
76         # keep result of each query inside the transaction
77         $result = mysqli_query($conn, "INSERT INTO supplied (contract_number,
78             supplied_product, supplied_amount, supplied_cost) VALUES (
79             {$contract_number}, '{$key}', {$amount}, {$cost})");
80
81         if (!$result) {
82             $failed = true;
83
84             # rollback the transaction if any query is failed
85             mysqli_query($conn, "ROLLBACK");
86             break;
87         }
88     }
89
90     if (!$failed) {
91         # commit the transaction if there are no failed queries
92         mysqli_query($conn, "COMMIT");
93     }
94 }

```

```

95      # restore autocommit property
96      mysqli_query($conn, "SET AUTOCOMMIT = 1");
97
98      # clear session array after products are stored into the database
99      $_SESSION["supplied_products"] = NULL;
100
101      header("location: index.php");
102  }
103  -?>

```

Розглянутий код файлу action.php необхідно доповнити наступним фрагментом, що призначений для формування та збереження документу Excel зі звітом про обсяги поставленої продукції за певний період. Для формування звіту буде використовуватися раніше створена збережена процедура «sp_contract_total».

Вміст файлу «manager.php» необхідно доповнити посиланням (рисунок 11), яке дозволить сформувати та завантажити звіт (рядок 21).

```

8      <h3>Contracts</h3>
9
10     <p>
11         # if the page is in record's create/update or delete mode (action parameter is set) - show 'back' link
12         if (isset($_GET["action"]) && ($_GET["action"] == "create" || $_GET["action"] == "update"
13             || $_GET["action"] == "delete")) {
14             ?>
15             <a href="index.php">Back</a>
16         </?php>
17         # otherwise - show 'new record' link
18         } else {
19             ?>
20             <a href="index.php?action=create">New contract</a>
21             <a href="index.php?action=export">Export data</a>
22         </?php>
23     }
24     ?>
25 </p>

```

Крім того, файл «action.php» необхідно доповнити кодом (рядки 104 – 127), призначеним безпосередньо для формування та завантаження звіту (рисунок 12).

```

104 # process request to export report into the Excel document
105 if (isset($_GET["action"]) && $_GET["action"] == "export") {
106     $filename = "report_contracts_" . date('Ymd') . ".xls";
107
108     header("Content-Disposition: attachment; filename=\"$filename\"");
109     header("Content-Type: application/vnd.ms-excel");
110
111     $flag = false;
112     $result = mysqli_query($conn, "CALL sp_contract_total('2018-01-01', CURRENT_TIMESTAMP())");
113
114     while ($row = mysqli_fetch_assoc($result)) {
115         if (!$flag) {
116             echo implode("\t", array_keys($row)) . "\r\n";
117             $flag = true;
118         }
119
120         array_walk($row, __NAMESPACE__ . '\cleanData');
121         echo implode("\t", array_values($row)) . "\r\n";
122     }
123
124     exit;
125 }
126
127 function cleanData(&$str) {
128     $str = preg_replace("/\t/", "\\t", $str);
129     $str = preg_replace("/\r?\n/", "\\n", $str);
130
131     if (strpos($str, "'") {
132         $str = "'" . str_replace("'", "", $str) . "'";
133     }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

Contracts

New contract Export data

Рисунок 11

A1		fx		contract_number	
	A	B	C	D	
1	contract_number	contract_date	SUM(supplied.supplied_amount)	SUM(supplied.supplied_amount * supplied.supplied_cost)	
2	1	9/1/2018 0:00	47	39500	
3	2	9/10/2018 0:00	24	11350	
4	3	9/23/2018 0:00	148	99600	
5	4	9/24/2018 0:00	119	76112.5	
6	5	10/2/2018 0:00	64	45630	
7	6	12/27/2018 13:30	15	59985	
8	13	1/10/2019 13:20			

Рисунок 12

Програмный код файла «action.php»:

<?php

process request to create contract

```

if (isset($_POST["create_contract"])) {
    $supplier_id = $_POST["supplier_id"];
    $contract_note = $_POST["contract_note"];

```

use the stored procedure created earlier

```

$sql = "CALL sp_contract_ops(i, 0, { $supplier_id },
{'$contract_note'})";

```

```

mysql_query($conn, $sql);

header("location: index.php");
}

# process request to delete contract
if (isset($_POST["delete_contract"])) {
    $contract_number = $_POST["contract_number"];

    $sql = "CALL sp_contract_ops('d', {$contract_number}, ", 0, ")";
    mysql_query($conn, $sql);

    header("location: index.php");
}

# process request to update contract
if (isset($_POST["update_contract"])) {
    $contract_number = $_POST["contract_number"];
    $supplier_id = $_POST["supplier_id"];
    $contract_note = $_POST["contract_note"];

    $sql = "CALL sp_contract_ops('u', {$contract_number},
CURRENT_TIMESTAMP(), {$supplier_id}, '{$contract_note}')";
    mysql_query($conn, $sql);

    header("location: index.php");
}

# process request to insert new record into session array of delivered
products
if (isset($_POST["add_product"])) {
    $supplied_product = $_POST["supplied_product"];
    $supplied_amount = $_POST["supplied_amount"];
    $supplied_cost = $_POST["supplied_cost"];

    if (!empty($supplied_product) && !empty($supplied_amount) && !empty(
$supplied_cost)) {
        if (is_numeric($supplied_amount) && is_numeric($supplied_cost)) {
            if ($supplied_amount > 0 && $supplied_cost > 0) {

                $_SESSION["supplied_products"][$supplied_product] = array("amount" =
> $supplied_amount,
                    "cost" => $supplied_cost);
            }

```

```

    }
}

header("location: index.php");
}

# process request to remove a record from the session array
if (isset($_GET["supplied"]) && $_GET["supplied"] == "remove") {
    $supplied_product = $_GET["product"];
    unset($_SESSION["supplied_products"][$supplied_product]);

    header("location: index.php");
}

# process request to store delivered products into the database
if (isset($_POST["save_products"])) {
    $contract_number = $_POST["contract_number"];

    # begin transaction
    mysqli_query($conn, "SET AUTOCOMMIT = 0");
    mysqli_query($conn, "START TRANSACTION");

    $failed = false;

    foreach ($_SESSION["supplied_products"] as $key => $value) {
        $amount = $value["amount"];
        $cost = $value["cost"];

        # keep result of each query inside the transaction
        $result = mysqli_query($conn, "INSERT INTO supplied
(contract_number,
    supplied_product, supplied_amount, supplied_cost) VALUES (
    {$contract_number}, '{$key}', {$amount}, {$cost})");

        if (!$result) {
            $failed = true;

            # rollback the transaction if any query is failed
            mysqli_query($conn, "ROLLBACK");
            break;
        }
    }

    if (!$failed) {

```

```

# commit the transaction if there are no failed queries
mysql_query($conn, "COMMIT");
}

# restore autocommit property
mysql_query($conn, "SET AUTOCOMMIT = 1");

# clear session array after products are stored into the database
$_SESSION["supplied_products"] = NULL;

header("location: index.php");
}

# process request to export report into the Excel document
if (isset($_GET["action"]) && $_GET["action"] == "export") {
    $filename = "report_contracts_" . date("Ymd") . ".xls";

    header("Content-Disposition: attachment; filename=\"$filename\"");
    header("Content-Type: application/vnd.ms-excel");
    $flag = false;
    $result = mysql_query($conn, "CALL sp_contract_total('2018-01-01',
CURRENT_TIMESTAMP())");

    while ($row = mysql_fetch_assoc($result)) {
        if (!$flag) {
            echo implode("\t", array_keys($row)) . "\r\n";
            $flag = true;
        }
        array_walk($row, __NAMESPACE__ . '\cleanData');
        echo implode("\t", array_values($row)) . "\r\n";
    }

    exit;
}

function cleanData(&$str) {
    $str = preg_replace("/\t/", "\\t", $str);
    $str = preg_replace("/\r?\n/", "\\n", $str);

    if (strstr($str, "'")) {
        $str = "'" . str_replace("'", "", $str) . "'";
    }
}
?>

```

6. ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ

1. Розробити функціональність програмного забезпечення для роботи адміністратора бази даних «supply». Адміністратор повинен мати можливість створювати, оновлювати та видаляти записи в усіх таблицях бази даних.

2. Доповнити розглянутий фрагмент програмного забезпечення для роботи з БД «supply» можливістю сортувати рядки у таблиці Contracts (файл «manager.php») в прямому та зворотному напрямках:

- за номером договору;
- за датою укладення договору.

3. Доповнити розглянутий фрагмент програмного забезпечення для роботи з БД «supply» можливістю сортувати рядки у таблиці «Supplied products by contract #X» (файл «manager.php») в прямому та зворотному напрямках:

- за назвою поставленого товару;
- за кількістю одиниць поставленого товару;
- за вартістю одиниці поставленого товару.

4. Під час спроби оновлення даних про певний договір відкривається відповідна форма, що містить елемент «combo box» зі списком постачальників. Внести зміни до програмного забезпечення таким чином, щоб у цьому «combo box» одразу після завантаження форми був обраний постачальник, з яким на даний момент укладений договір, що редагується.

5. Зважаючи на механізм посилкової цілісності, що використовується у базі даних «supply», наразі неможливим є видалення даних про договір, за яким були поставлені товари. Внести зміни до програмного забезпечення (наприклад, до збереженої процедури «sp_contract_ops») для того, щоб дані про договір можна було видалити незважаючи на наявність даних про поставлені за цим договором товари.

6. Зважаючи на механізм посилкової цілісності, що використовується у базі даних «supply», наразі неможливим є видалення даних про договір, за яким були поставлені товари. Внести зміни до програмного забезпечення таким чином, що для не «порожніх» договорів не буде доступною спроба їх видалення.

7. Як видно з рисунку 12, назви стовпців у згенерованому звіті не є зрозумілими для користувача, особливо це стосується стовпців, що

містять агреговані дані. Внести зміни до програмного забезпечення таким чином, щоб вказати назви «Contract», «Date», «Total amount» та «Total cost» для відповідних стовпців.

8. Поточна реалізація дозволяє сформувати звіт (рисунок 12) на основі фіксованого діапазону дат – починаючи з 1 січня 2018 року до часу генерації звіту. Внести зміни до програмного забезпечення таким чином, щоб користувач мав змогу самостійно вказувати необхідний період для генерації звіту про обсяги поставленої продукції.

9. Надати менеджеру з закупівель можливість працювати з даними про постачальників (додавати записи, оновлювати та видаляти існуючі записи). Забезпечити можливість перегляду переліку договорів, укладених з певним постачальником.

10. Доповнити програмне забезпечення автоматичною генерацією прибуткової накладної відразу після того, як дані про товари, що надійшли від постачальника в рамках певного договору на поставку, були збережені працівником складу до бази даних.