

ДОДАТОК А

Лістинг програмного коду

Код файлу «ajax_handler.php»:

```

<?php
include 'config.php';
check_auth(1); // ТІЛЬКИ вчитель
$teacher_id = $_SESSION['user_id'];
if (!isset($_POST['action'])) die('Invalid request');
$action = $_POST['action'];

// Дія додавання питання
if ($action == 'add_question') {
    $quiz_id = (int)$_POST['quiz_id'];
    $text = $_POST['question_text'];
    $points = (int)$_POST['points'];
    $type = $_POST['type'];
    if ($points < 1) $points = 1;

    // Запис питання в БД
    $stmt = $db->prepare("INSERT INTO questions (quiz_id, question_text, points, type)
VALUES (?, ?, ?, ?)");
    $stmt->bind_param("isis", $quiz_id, $text, $points, $type);
    $stmt->execute();
    $new_id = $db->insert_id;
    $typeLabel = 'Одна відповідь';
    if ($type == 'multiple') $typeLabel = 'Де кілька відповідей';
    if ($type == 'text') $typeLabel = 'Текстова відповідь';

    // Генеруємо HTML блок для нового питання і повертаємо його
    echo '
<div class="question-block" id="question-'. $new_id.'" data-type="'. $type.'">
    <div class="question-header">
        <span>
            <strong>'. htmlspecialchars($text).'</strong>
            <span class="question-meta">'. $points.' б.</span>
            <span class="question-meta" style="background:#d1ecf1;
color:#0c5460;">'. $typeLabel.'</span>
        </span>
        <button class="btn btn-delete" onclick="deleteItem(\'question\',
'. $new_id.')">Видалити</button>
    </div>';

    // Якщо це не текстове питання, додаємо блок для відповідей
    if ($type != 'text') {
        echo '<ul class="answers-list" id="answers-for-'. $new_id.'"></ul>
        <form class="add-answer-form" style="background: #f9f9f9; padding: 5px;"
onsubmit="addAnswer(event, \''. $new_id.')">
            <input type="text" name="answer_text" placeholder="Варіант відповіді" required
style="flex-grow:1;">
            <label><input type="checkbox" name="is_correct"> Правильна</label>
    
```

```

        <button type="submit" class="btn btn-add" style="padding: 4px 8px;">+</button>
    </form>;
} else {
    echo '<div style="padding:10px; color:#777; font-style:italic;">Це питання відкритого
типу. Учень вводитиме відповідь вручну.</div>';
}
echo '</div>';
exit;
}

// Дія додати відповідь на питання
if ($action == 'add_answer') {
    $q_id = (int)$_POST['question_id'];
    $text = $_POST['answer_text'];
    $is_correct = (int)$_POST['is_correct'];
    $q_type = $_POST['question_type'];
    // Якщо питання з однією відповіддю і ми додаємо правильну відповідь, то всі інші
відповіді цього питання стають неправильними
    if ($q_type == 'single' && $is_correct == 1) {
        $db->query("UPDATE answers SET is_correct = 0 WHERE question_id = $q_id");
    }

    $stmt = $db->prepare("INSERT INTO answers (question_id, answer_text, is_correct)
VALUES (?, ?, ?)");
    $stmt->bind_param("isi", $q_id, $text, $is_correct);
    $stmt->execute();
    $new_id = $db->insert_id;
    $class = $is_correct ? 'correct' : "";
    $check = $is_correct ? '✓' : "";
    echo '<li id="answer-'. $new_id. '" class="'. $class. '">
        <span>'. htmlspecialchars($text). ' '. $check. '</span>
        <button class="btn btn-delete" style="padding: 2px 5px; font-size: 12px;"
onclick="deleteItem(\'answer\', \''. $new_id. '\")>x</button>
        </li>';
    exit;
}

// Дія видалення
if ($action == 'delete_question') {
    $id = (int)$_POST['id'];
    $db->query("DELETE FROM questions WHERE id = $id");
    echo 'success';
    exit;
}
if ($action == 'delete_answer') {
    $id = (int)$_POST['id'];
    $db->query("DELETE FROM answers WHERE id = $id");
    echo 'success';
    exit;
}
?>

```

Код файлу «config.php»:

```

<?php
// Початок сесії на всіх сторінках
session_start();

// Налаштування Бази Даних
define('DB_HOST', 'localhost');
define('DB_USER', 'root'); // логін
define('DB_PASS', 'root'); // пароль
define('DB_NAME', 'student_success_db');

// Створення підключення
$db = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);

// Перевірка підключення
if ($db->connect_error) {
    die("Помилка підключення до MySQL: " . $db->connect_error);
}

// Встановлення кодування
$db->set_charset("utf8mb4");

/**
 * Функція для перевірки авторизації та ролі
 * @param int $required_role 1=Вчитель, 2=Учень, 3=Адмін
 */
function check_auth($required_role) {
    // Якщо не авторизований або немає ролі в сесії
    if (!isset($_SESSION['user_id']) || !isset($_SESSION['role'])) {
        header('Location: index.php');
        exit;
    }

    // Якщо роль не співпадає з необхідною на цій сторінці
    if ($_SESSION['role'] != $required_role) {
        // Перенаправлення користувачів
        if ($_SESSION['role'] == 1) {
            header('Location: teacher_quizzes.php');
        } elseif ($_SESSION['role'] == 2) {
            header('Location: student_dashboard.php');
        } elseif ($_SESSION['role'] == 3) {
            header('Location: admin_dashboard.php');
        }
        exit;
    }
}
?>

```

Код файлу «create_quiz.php»:

```

<?php
include 'config.php';
check_auth(1);
$teacher_id = $_SESSION['user_id'];
$subject_id = $_SESSION['teacher_subject_id'];

// Перевірка, чи призначено вчителю предмет
if (!$subject_id) {
    die("Помилка: Вашому акаунту не призначено предмет. Зверніться до адміністратора.");
}

// Отримуємо назву предмету для відображення на сторінці
$sub_query = $db->query("SELECT name FROM subjects WHERE id = $subject_id");
$subject_name = $sub_query->fetch_assoc()['name'];

// Обробка форми
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $title = trim($_POST['title']);
    $class_name = $_POST['class_name'];
    $topic_name = trim($_POST['topic_name']);
    $time_limit = (int)$_POST['time_limit'];

    if (empty($topic_name) || empty($title)) {
        die("Будь ласка, заповніть всі поля.");
    }

    // Логіка роботи з темою
    $check_topic = $db->prepare("SELECT id FROM topics WHERE topic_name = ? AND
subject_id = ?");
    $check_topic->bind_param("si", $topic_name, $subject_id);
    $check_topic->execute();
    $topic_res = $check_topic->get_result();

    if ($topic_res->num_rows > 0) {
        $topic_row = $topic_res->fetch_assoc();
        $topic_id = $topic_row['id'];
    } else {
        $insert_topic = $db->prepare("INSERT INTO topics (subject_id, topic_name) VALUES (?,
?");
        $insert_topic->bind_param("is", $subject_id, $topic_name);
        $insert_topic->execute();
        $topic_id = $db->insert_id;
        $insert_topic->close();
    }
    $check_topic->close();

    // Створення тестів
    $stmt = $db->prepare("INSERT INTO quizzes (teacher_id, topic_id, title, class_name,
time_limit) VALUES (?, ?, ?, ?, ?");
    $stmt->bind_param("iiss", $teacher_id, $topic_id, $title, $class_name, $time_limit);

```

```

if ($stmt->execute()) {
    $new_quiz_id = $db->insert_id;
    header('Location: edit_quiz.php?id=' . $new_quiz_id);
    exit;
} else {
    echo "Помилка створення тесту: " . $db->error;
}
$stmt->close();
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Створити Новий Тест</title>
    <style>
        body { font-family: Arial, sans-serif; display: grid; place-items: center; min-height: 100vh;
background-color: #f4f4f4; margin: 0; }

        form {
            background: #fff;
            border: 1px solid #ccc;
            padding: 25px;
            border-radius: 8px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
            min-width: 400px;
        }

        h2 {
            text-align: center;
            margin-top: 0;
            color: #333;
        }

        div { margin-bottom: 15px; }

        label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }

        input[type="text"], input[type="number"], select {
            width: 100%;
            padding: 8px;
            border: 1px solid #ddd;
            border-radius: 4px;
            box-sizing: border-box;
        }

        button {
            width: 100%;
            padding: 10px;
            background-color: #28a745;
            color: white;

```

```

border: none;
border-radius: 4px;
cursor: pointer;
font-size: 16px;
margin-top: 10px;
}
button:hover { background-color: #218838; }

.btn-cancel {
display: block;
width: 100%;
padding: 10px;
background-color: #007bff;
color: white;
text-decoration: none;
border-radius: 4px;
text-align: center;
font-size: 16px;
margin-top: 10px;
box-sizing: border-box;
}
.btn-cancel:hover { background-color: #0056b3; }

.subject-info {
background-color: #e3f2fd;
color: #0d47a1;
padding: 10px;
border-radius: 5px;
margin-bottom: 20px;
text-align: center;
border: 1px solid #90caf9;
}
</style>
</head>
<body>
<form action="create_quiz.php" method="POST">
<h2>Створення Тесту</h2>

<div class="subject-info">
Ви створюєте тест з предмету: <strong><?php echo htmlspecialchars($subject_name);
?></strong>
</div>

<div>
<label for="title">Назва тесту:</label>
<input type="text" id="title" name="title" required>
</div>

<div>
<label for="class_name">Для якого класу/групи:</label>
<select id="class_name" name="class_name" required>
<option value="">-- Оберіть клас --</option>

```

```

<?php
$class_sql = "SELECT DISTINCT class_name FROM users WHERE role = 2 AND
class_name IS NOT NULL ORDER BY class_name";
$class_res = $db->query($class_sql);
while ($row = $class_res->fetch_assoc()) {
    echo "<option value=\"" . htmlspecialchars($row['class_name']) . "\"" .
htmlspecialchars($row['class_name']) . "</option>";
}
?>
</select>
</div>

<div>
<label for="topic_name">Тема уроку/тесту:</label>
<input type="text" id="topic_name" name="topic_name" required>
</div>

<div>
<label for="time_limit">Ліміт часу (у хвиликах):</label>
<input type="number" id="time_limit" name="time_limit" min="1" value="15"
required>
</div>

<button type="submit">Створити та перейти до питань</button>
<a href="teacher_quizzes.php" class="btn-cancel">Скасувати</a>
</form>
</body>
</html>

```

Код файлу «edit_quiz.php»:

```

<?php
include 'config.php';
// Перевірка (доступ тільки для вчителя) роль 1
check_auth(1);
$teacher_id = $_SESSION['user_id'];

// Якщо ID тесту не передано то повертаємо на список
if (!isset($_GET['id'])) {
    header('Location: teacher_quizzes.php');
    exit;
}
$quiz_id = (int)$_GET['id'];

// Обробка форми налаштувань
if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST['update_settings'])) {
    $new_title = $_POST['title'];
    $new_time = (int)$_POST['time_limit'];

    // Оновлення заголовку і часу у БД
    $upd_stmt = $db->prepare("UPDATE quizzes SET title = ?, time_limit = ? WHERE id = ?
AND teacher_id = ?");

```

```

$upd_stmt->bind_param("siii", $new_title, $new_time, $quiz_id, $teacher_id);
$upd_stmt->execute();
$upd_stmt->close();

// Перезавантаження сторінки
header("Location: edit_quiz.php?id=$quiz_id");
exit;
}

// Отримання даних про тест
$quiz_stmt = $db->prepare("SELECT title, time_limit FROM quizzes WHERE id = ? AND
teacher_id = ?");
$quiz_stmt->bind_param("ii", $quiz_id, $teacher_id);
$quiz_stmt->execute();
$quiz = $quiz_stmt->get_result()->fetch_assoc();
if (!$quiz) die("Тест не знайдено.");
$quiz_stmt->close();
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Редактор Тесту</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4; }
    header { display: flex; justify-content: space-between; align-items: center; padding-bottom:
10px; border-bottom: 2px solid #ccc; margin-bottom: 20px; background-color: #fff; padding:
15px 20px; border-radius: 5px; box-shadow: 0 2px 5px rgba(0,0,0,0.05); }
    header h1 { margin: 0; font-size: 24px; }
    .back-btn { text-decoration: none; color: white; background-color: #007bff; padding: 8px
16px; border-radius: 4px; font-weight: bold; font-size: 14px; transition: background-color 0.2s; }
    .back-btn:hover { background-color: #0056b3; }
    .settings-box { background: #fff; padding: 15px; border: 1px solid #ccc; border-radius: 5px;
margin-bottom: 20px; }
    .settings-box input { padding: 5px; margin-right: 10px; }
    .question-block { background: #fff; border: 1px solid #ddd; border-radius: 5px; padding:
15px; margin-bottom: 15px; box-shadow: 0 1px 3px rgba(0,0,0,0.1); }
    .question-header { display: flex; justify-content: space-between; align-items: center; border-
bottom: 1px solid #eee; padding-bottom: 10px; margin-bottom: 10px; }
    .question-meta { font-size: 0.9em; color: #666; background: #eee; padding: 2px 6px;
border-radius: 4px; margin-left: 10px; }
    .btn { padding: 5px 10px; cursor: pointer; border: none; border-radius: 4px; color: white; }
    .btn-save { background: #28a745; }
    .btn-delete { background: #dc3545; }
    .btn-add { background: #007bff; }
    .add-question-form { background: #e3f2fd; padding: 15px; border-radius: 5px; display:
flex; gap: 10px; align-items: center; }
    .add-question-form input, .add-question-form select { padding: 8px; }
    .answers-list { list-style: none; padding-left: 0; }
    .answers-list li { display: flex; justify-content: space-between; padding: 5px; border-bottom:
1px solid #f0f0f0; }
    .answers-list .correct { color: green; font-weight: bold; }

```

```

</style>
</head>
<body>
  <header>
    <h1>Редактор: <?php echo htmlspecialchars($quiz['title']); ?></h1>
    <a href="teacher_quizzes.php" class="back-btn">← Назад до списку</a>
  </header>

  <div class="settings-box">
    <form method="POST">
      <label>Назва:</label> <input type="text" name="title" value="<?php echo
htmlspecialchars($quiz['title']); ?>" required style="width: 300px;">
      <label>Час (хв):</label> <input type="number" name="time_limit" value="<?php echo
$quiz['time_limit']; ?>" min="1" required>
      <button type="submit" name="update_settings" class="btn btn-
save">Зберегти</button>
    </form>
  </div>

  <div id="quiz-questions">
    <?php
    // Завантаження всіх питань з цього тесту
    $q_stmt = $db->prepare("SELECT * FROM questions WHERE quiz_id = ? ORDER BY id
ASC");
    $q_stmt->bind_param("i", $quiz_id);
    $q_stmt->execute();
    $questions = $q_stmt->get_result();

    while ($q = $questions->fetch_assoc()):
      // Визначення текстової мітки для типу питання
      $titleLabel = 'Одна відповідь';
      if ($q['type'] == 'multiple') $titleLabel = 'Декілька відповідей';
      if ($q['type'] == 'text') $titleLabel = 'Текстова відповідь';
    ?>
    <div class="question-block" id="question-<?php echo $q['id']; ?>" data-type="<?php
echo $q['type']; ?>">
      <div class="question-header">
        <span>
          <strong><?php echo htmlspecialchars($q['question_text']); ?></strong>
          <span class="question-meta"><?php echo $q['points']; ?> б.</span>
          <span class="question-meta" style="background:#d1ecf1;
color:#0c5460;"><?php echo $titleLabel; ?></span>
        </span>
        <button class="btn btn-delete" onclick="deleteItem('question', <?php echo $q['id'];
?>)">Видалити</button>
      </div>

      <?php if ($q['type'] != 'text'): ?>
        <ul class="answers-list" id="answers-for-<?php echo $q['id']; ?>">
          <?php
          $a_stmt = $db->prepare("SELECT * FROM answers WHERE question_id = ?
ORDER BY id ASC");

```

```

    $_stmt->bind_param("i", $q['id']);
    $_stmt->execute();
    $answers = $_stmt->get_result();
    while ($a = $answers->fetch_assoc()):
    ?>
        <li id="answer-<?php echo $a['id']; ?>" class="<?php echo $a['is_correct'] ?
'correct' : ''; ?>">
            <span><?php echo htmlspecialchars($a['answer_text']); ?> <?php echo
$a['is_correct'] ? '✓' : ''; ?></span>
            <button class="btn btn-delete" style="padding: 2px 5px; font-size: 12px;"
onclick="deleteItem('answer', <?php echo $a['id']; ?>)">x</button>
            </li>
        <?php endwhile; $_stmt->close(); ?>
    </ul>
    <form class="add-answer-form" style="background: #f9f9f9; padding: 5px;"
onsubmit="addAnswer(event, <?php echo $q['id']; ?>)">
        <input type="text" name="answer_text" placeholder="Варіант відповіді"
required style="flex-grow: 1;">
        <label><input type="checkbox" name="is_correct"> Правильна</label>
        <button type="submit" class="btn btn-add" style="padding: 4px
8px;">+</button>
    </form>
    <?php else: ?>
        <div style="padding: 10px; color: #777; font-style: italic;">Це питання відкритого
типу. Учень вводитиме відповідь вручну.</div>
    <?php endif; ?>
</div>
<?php endwhile; $q_stmt->close(); ?>
</div>

<div class="add-question-form">
    <select id="new-question-type">
        <option value="single">Одна правильна відповідь (Тест)</option>
        <option value="multiple">Декілька правильних відповідей (Тест)</option>
        <option value="text">Текстова відповідь (Ручна перевірка)</option>
    </select>
    <input type="text" id="new-question-text" placeholder="Текст нового питання"
style="flex-grow: 1;">
    <input type="number" id="new-question-points" value="1" min="1" placeholder="Бали"
style="width: 60px;">
    <button class="btn btn-add" onclick="addQuestion()">Додати Питання</button>
</div>

<script>
    const QUIZ_ID = <?php echo $quiz_id; ?>;

    // Універсальна функція для відправки AJAX запитів
    async function sendAjax(data) {
        try {
            const response = await fetch('ajax_handler.php', {
                method: 'POST',
                headers: { 'Content-Type': 'application/x-www-form-urlencoded' },

```

```

        body: new URLSearchParams(data)
    });
    return await response.text();
} catch (error) { console.error(error); }
}

// Додавання питання
async function addQuestion() {
    const textInput = document.getElementById('new-question-text');
    const pointsInput = document.getElementById('new-question-points');
    const typeInput = document.getElementById('new-question-type');

    if (textInput.value.trim() === "") return;

    // Відправка даних на сервер
    const responseHtml = await sendAjax({
        action: 'add_question',
        quiz_id: QUIZ_ID,
        question_text: textInput.value.trim(),
        points: pointsInput.value,
        type: typeInput.value
    });

    // Додавання отриманого HTML в список
    if (responseHtml) {
        document.getElementById('quiz-questions').innerHTML += responseHtml;
        textInput.value = "";
        pointsInput.value = "1";
    }
}

// Додавання відповіді
async function addAnswer(event, questionId) {
    event.preventDefault();
    const form = event.target;
    const answerText = form.querySelector('input[name="answer_text"]').value.trim();
    const isCorrect = form.querySelector('input[name="is_correct"]').checked ? 1 : 0;

    // Отримання типу питання
    const questionBlock = document.getElementById('question-' + questionId);
    const qType = questionBlock.getAttribute('data-type');

    if (answerText === "") return;

    const responseHtml = await sendAjax({
        action: 'add_answer',
        question_id: questionId,
        answer_text: answerText,
        is_correct: isCorrect,
        question_type: qType
    });
}

```

```

    if (responseHtml) {
        // Якщо це питання з однією правильною відповіддю і ми додали правильну, то
        знімаємо "галочку" з усіх інших варіантів відповіді
        if (qType === 'single' && isCorrect) {
            const allAnswers = document.querySelectorAll(`#answers-for-${questionId} li`);
            allAnswers.forEach(li => li.classList.remove('correct'));
        }
        document.getElementById(`answers-for-${questionId}`).innerHTML +=
responseHtml;
        form.reset();
    }
}

// Видалення
async function deleteItem(type, id) {
    if (confirm('Видалити?')) {
        const response = await sendAjax({ action: (type === 'question' ? 'delete_question' :
'delete_answer'), id: id });
        if (response === 'success') {
            document.getElementById(`${type}-${id}`).remove();
        } else {
            alert('Помилка видалення');
        }
    }
}
</script>
</body>
</html>

```

Код файлу «index.php»:

```

<?php
include 'config.php';

// Якщо користувач вже увійшов перенаправляємо його
if (isset($_SESSION['user_id'])) {
    if ($_SESSION['role'] == 1) {
        header('Location: teacher_dashboard.php');
    } else {
        header('Location: student_dashboard.php');
    }
    exit;
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Вхід - Система Успішності</title>
    <style>
        /* Загальний стиль сторінки */
    body {

```

```
font-family: Arial, sans-serif;
margin: 0;
height: 100vh;
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;

/* Фон */
background-color: #f4f4f4;
background-image: url('background.jpg');
background-size: cover;
background-position: center;
background-repeat: no-repeat;
}

/* Привітання */
.welcome-container {
background-color: rgba(255, 255, 255, 0.95);
padding: 20px 30px;
border-radius: 5px;
box-shadow: 0 4px 15px rgba(0,0,0,0.2);
width: 400px;
text-align: center;
margin-bottom: 20px;
border: 1px solid #ccc;
}

.welcome-container h1 {
margin-top: 0;
font-size: 22px;
color: #1f8a1fff;
margin-bottom: 10px;
}

.welcome-container p {
margin: 0;
font-size: 16px;
color: #555;
line-height: 1.5;
}

/* Стиль вікна авторизації */
.login-container {
background-color: rgba(255, 255, 255, 0.95);
padding: 30px;
border-radius: 5px;
box-shadow: 0 4px 15px rgba(0,0,0,0.2);
width: 350px;
border: 1px solid #ccc;
}
```

```
/* Заголовок формы */
h2 {
  text-align: center;
  color: #333;
  margin-top: 0;
  margin-bottom: 25px;
  font-size: 24px;
}

/* Поля ввода */
.form-group {
  margin-bottom: 15px;
}

label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
  color: #333;
  font-size: 14px;
}

input[type="text"],
input[type="password"] {
  width: 100%;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
  font-size: 14px;
}

input:focus {
  border-color: #007bff;
  outline: none;
}

/* Кнопка */
button {
  width: 100%;
  padding: 12px;
  background-color: #3062ceff;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
  font-weight: bold;
  margin-top: 10px;
  transition: background-color 0.2s;
}
```

```

button:hover {
  background-color: #13961dff;
}

/* Повідомлення про помилку */
.error {
  background-color: #f8d7da;
  color: #721c24;
  border: 1px solid #f5c6cb;
  padding: 10px;
  border-radius: 4px;
  margin-bottom: 20px;
  text-align: center;
  font-size: 14px;
}
</style>
</head>
<body>

  <div class="welcome-container">
    <h1>Вітаємо в системі перевірки та аналізу успішності!</h1>
    <p>Авторизуйтеся у системі за допомогою логіна та паролю як вчитель (викладач),
або учень (студент).</p>
  </div>

  <div class="login-container">
    <h2>Авторизація</h2>

    <?php if (isset($_GET['error'])): ?>
      <div class="error">Неправильний логін або пароль.</div>
    <?php endif; ?>

    <form action="login_handler.php" method="POST">
      <div class="form-group">
        <label for="login">Логін:</label>
        <input type="text" id="login" name="login" required autofocus>
      </div>
      <div class="form-group">
        <label for="password">Пароль:</label>
        <input type="password" id="password" name="password" required>
      </div>
      <button type="submit">Увійти</button>
    </form>
  </div>

</body>
</html>

```

Код файлу «login_handler.php»:

```

<?php
include 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $login = $_POST['login'];
    $password = $_POST['password'];

    // Отримуємо дані користувача
    $stmt = $db->prepare("SELECT id, name, password, role, subject_id FROM users WHERE
login = ?");
    $stmt->bind_param("s", $login);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 1) {
        $user = $result->fetch_assoc();

        // Перевірка хешу пароля
        if (password_verify($password, $user['password'])) {
            // Записуємо дані в сесію
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['name'] = $user['name'];
            $_SESSION['role'] = $user['role'];
            $_SESSION['teacher_subject_id'] = $user['subject_id']; // Для вчителів

            // Перенаправлення користувачів на їх сторінки
            if ($user['role'] == 1) {
                header('Location: teacher_quizzes.php');
            } elseif ($user['role'] == 2) {
                header('Location: student_dashboard.php');
            } elseif ($user['role'] == 3) {
                // Панель адміна
                header('Location: admin_dashboard.php');
            }
            exit;
        }
    }
    // Якщо помилка
    header('Location: index.php?error=1');
    exit;
} else {
    header('Location: index.php');
    exit;
}
?>

```

Код файлу «logout.php»:

```

<?php
include 'config.php';

// Закриваємо сесію
session_unset();
session_destroy();

// Перенаправлення на сторінку входу
header('Location: index.php');
exit;
?>

```

Код файлу «predeiction.php»:

```

<?php

function predictNextScore(array $scores) {
    $n = count($scores);

    // Мінімум 3 тесту для прогнозу
    if ($n < 3) {
        return null;
    }

    $sum_x = 0;
    $sum_y = 0;
    $sum_xy = 0;
    $sum_xx = 0;

    // x = порядковий номер, y = оцінка
    for ($i = 0; $i < $n; $i++) {
        $x = $i + 1;
        $y = (float)$scores[$i];

        $sum_x += $x;
        $sum_y += $y;
        $sum_xy += ($x * $y);
        $sum_xx += ($x * $x);
    }

    $denominator_m = ($n * $sum_xx) - ($sum_x * $sum_x);

    if ($denominator_m == 0) {
        // Якщо всі x однакові або помилка обчислень
        return round(array_sum($scores) / $n, 2); // Повертаємо середнє
    }
}

```

```

$m = (($n * $sum_xy) - ($sum_x * $sum_y)) / $denominator_m;
$b = ($sum_y - $m * $sum_x) / $n;

// Прогнозуємо для наступного кроку (x = n + 1)
$predicted_score = ($m * ($n + 1)) + $b;

// Обмеження 0-100%
if ($predicted_score > 100) $predicted_score = 100.0;
if ($predicted_score < 0) $predicted_score = 0.0;

return round($predicted_score, 2);
}
?>

```

Код файлу «student_dashboard.php»:

```

<?php
include 'config.php';
check_auth(2); // Тільки для учнів
$student_id = $_SESSION['user_id'];

// Отримуємо клас студента
$user_query = $db->query("SELECT class_name FROM users WHERE id = $student_id");
$student_class = $user_query->fetch_assoc()['class_name'];

// Отримуємо обраний предмет для фільтрації
$filter_subject_id = isset($_GET['subject_id']) ? (int)$_GET['subject_id'] : 0;
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Мої Тести</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
      background-color: #f4f4f4;
      color: #333;
    }

    /* header */
    header {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding-bottom: 10px;
      border-bottom: 2px solid #ccc;
      margin-bottom: 20px;
      background-color: #fff;
      padding: 15px 20px;
    }

```

```

    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.05);
}
header h1 { margin: 0; font-size: 24px; }

/* СТИЛЬ КНОПОК МЕНЮ */
nav { display: flex; gap: 10px; }

nav a {
    text-decoration: none;
    color: white;
    background-color: #6c757d;
    padding: 8px 16px;
    border-radius: 4px;
    font-weight: bold;
    font-size: 14px;
    transition: background-color 0.2s, transform 0.1s;
}

nav a:hover {
    background-color: #5a6268;
    text-decoration: none;
}

/* СТИЛЬ КНОПКИ ПОТОЧНОЇ СТОРІНКИ */
nav a.active {
    background-color: #495057;
    cursor: default;
    box-shadow: inset 0 2px 4px rgba(0,0,0,0.2);
}

/* СТИЛЬ КНОПКИ ВИЙТИ */
nav a[href="logout.php"] {
    background-color: #dc3545;
}
nav a[href="logout.php"]:hover {
    background-color: #c82333;
}

.class-info { color: #777; font-size: 0.9em; margin-top: 5px; display: block;}

/* Фільтр */
.filter-container {
    background: #fff;
    padding: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-bottom: 20px;
    display: flex;
    align-items: center;
    gap: 10px;
}

```

```
.filter-container select {
  padding: 8px;
  border-radius: 4px;
  border: 1px solid #ccc;
  font-size: 14px;
  min-width: 200px;
}

/* Таблиця */
.table-container {
  background: #fff;
  border: 1px solid #ccc;
  border-radius: 5px;
  overflow: hidden;
}

table { width: 100%; border-collapse: collapse; }
th, td {
  padding: 15px 20px;
  text-align: left;
  border-bottom: 1px solid #eee;
  vertical-align: middle;
}
th { background-color: #f8f9fa; color: #333; font-weight: bold; }
tr:last-child td { border-bottom: none; }

td.action-cell { text-align: center; }

/* Кнопки дій */
.btn-take {
  background-color: #28a745;
  color: white;
  padding: 8px 15px;
  text-decoration: none;
  border-radius: 4px;
  font-size: 0.9em;
  font-weight: bold;
  display: inline-block;
}
.btn-take:hover { background-color: #218838; }

.btn-view {
  background-color: #007bff;
  color: white;
  padding: 8px 12px;
  text-decoration: none;
  border-radius: 4px;
  font-size: 0.85em;
  display: inline-block;
  margin-top: 10px;
}
.btn-view:hover { background-color: #0056b3; }
```

```

.result-block { display: flex; flex-direction: column; align-items: center; gap: 2px; }
.score-info { font-weight: bold; color: #2c3e50; font-size: 1.1em; }
.score-percent { color: #7f8c8d; font-size: 0.9em; }
.no-data { text-align: center; padding: 30px; background: #fff; border: 1px solid #ccc;
border-radius: 5px; color: #777; }
</style>
</head>
<body>
  <header>
    <div>
      <h1>Мої Тести</h1>
      <span class="class-info">Вітаємо, <?php echo htmlspecialchars($_SESSION['name']);
?> (Клас: <?php echo htmlspecialchars($student_class); ?>)</span>
    </div>
    <nav>
      <a href="student_dashboard.php" class="active">Мої Тести</a>
      <a href="student_grades.php">Аналітика</a>
      <a href="logout.php">Вийти</a>
    </nav>
  </header>

  <main>
    <div class="filter-container">
      <label for="subject_filter"><strong>Фільтрувати за предметом:</strong></label>
      <form action="student_dashboard.php" method="GET" style="margin: 0;">
        <select name="subject_id" id="subject_filter" onchange="this.form.submit()">
          <option value="0">-- Всі предмети --</option>
          <?php
            $subjects_res = $db->query("SELECT * FROM subjects ORDER BY name");
            while ($sub = $subjects_res->fetch_assoc()) {
              $selected = ($filter_subject_id == $sub['id']) ? 'selected' : '';
              echo "<option value='" . $sub['id'] . "' $selected>" .
htmlspecialchars($sub['name']) . "</option>";
            }
          ?>
        </select>
      </form>
    </div>

    <div class="table-container">
      <table>
        <thead>
          <tr>
            <th width="20%">Предмет</th>
            <th width="25%">Тема</th>
            <th width="30%">Назва Тесту</th>
            <th width="25%" style="text-align: center;">Дія / Результат</th>
          </tr>
        </thead>
        <tbody>
          <?php

```

```

$ssql = "SELECT
        q.id, q.title, t.topic_name, s.name AS subject_name,
        qr.score, qr.id AS result_id,
        (SELECT SUM(p.points) FROM questions p WHERE p.quiz_id = q.id) AS
max_points
        FROM quizzes q
        JOIN topics t ON q.topic_id = t.id
        JOIN subjects s ON t.subject_id = s.id
        LEFT JOIN quiz_results qr ON qr.quiz_id = q.id AND qr.student_id = ?
        WHERE q.class_name = ?";

if ($filter_subject_id > 0) {
    $ssql .= " AND s.id = ?";
}

$ssql .= " ORDER BY s.name, t.topic_name";

$stmt = $db->prepare($ssql);

if ($filter_subject_id > 0) {
    $stmt->bind_param("isi", $student_id, $student_class, $filter_subject_id);
} else {
    $stmt->bind_param("is", $student_id, $student_class);
}

$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . htmlspecialchars($row['subject_name']) . "</td>";
        echo "<td>" . htmlspecialchars($row['topic_name']) . "</td>";
        echo "<td>" . htmlspecialchars($row['title']) . "</td>";
        echo "<td class='action-cell'>";

        if ($row['score'] !== null) {
            $max = $row['max_points'] ? $row['max_points'] : 0;
            $earned = ($max > 0) ? round(($row['score'] / 100) * $max, 1) : 0;

            echo "<div class='result-block'>";
            echo "<div class='score-info'>{$earned} / {$max} балів</div>";
            echo "<div class='score-percent'>({$row['score']}%)</div>";
            echo "</div>";
            echo "<a href='view_result.php?id=" . $row['id'] . "' class='btn-
view'>Переглянути результати</a>";
        } else {
            echo "<a href='take_test.php?id=" . $row['id'] . "' class='btn-take'>Пройти
тест</a>";
        }
        echo "</td>";
    }
    echo "</tr>";
}

```

```

        }
    } else {
        echo "<tr><td colspan='4' style='text-align:center; padding: 30px;
color:#777;'>Тестів не знайдено.</td></tr>";
    }
    $stmt->close();
    ?>
</tbody>
</table>
</div>
</main>
</body>
</html>

```

Код файлу «student_grades.php»:

```

<?php
include 'config.php';
include 'prediction.php';
check_auth(2);
$student_id = $_SESSION['user_id'];

// Отримуємо клас студента
$user_query = $db->query("SELECT class_name FROM users WHERE id = $student_id");
$student_class = $user_query->fetch_assoc()['class_name'];

$selected_subject_id = isset($_GET['subject_id']) ? (int)$_GET['subject_id'] : 0;
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Аналіз Успішності</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            background-color: #f4f4f4;
            color: #333;
        }

        /* Шапка */
        header {
            display: flex;
            justify-content: space-between;
            align-items: center;
            padding-bottom: 10px;
            border-bottom: 2px solid #ccc;
            margin-bottom: 20px;
            background-color: #fff;
            padding: 15px 20px;

```

```

    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.05);
}
header h1 { margin: 0; font-size: 24px; }

.class-info { color: #777; font-size: 0.9em; margin-top: 5px; display: block;}

/* СТИЛЬ КНОПОК МЕНЮ */
nav { display: flex; gap: 10px; }

nav a {
    text-decoration: none;
    color: white;
    background-color: #6c757d;
    padding: 8px 16px;
    border-radius: 4px;
    font-weight: bold;
    font-size: 14px;
    transition: background-color 0.2s;
}

nav a:hover { background-color: #5a6268; }

nav a.active {
    background-color: #495057;
    cursor: default;
    box-shadow: inset 0 2px 4px rgba(0,0,0,0.2);
}

nav a[href="logout.php"] { background-color: #dc3545; }
nav a[href="logout.php"]:hover { background-color: #c82333; }

/* ФІЛЬТР */
.filter-section {
    background: #fff; padding: 15px; border: 1px solid #ccc;
    border-radius: 5px; margin-bottom: 20px;
}

select { padding: 8px; width: 300px; border: 1px solid #ccc; border-radius: 4px; }

/* Статистика */
.summary-card {
    background-color: #fff; border: 1px solid #ccc; padding: 20px;
    border-radius: 5px; margin-bottom: 20px; display: flex;
    justify-content: space-around; text-align: center;
}

.summary-item h3 { margin: 5px 0; font-size: 2em; color: #007bff; }
.summary-item p { margin: 0; color: #555; font-size: 1em; }

/* Блок теми */
.topic-block {
    background: #fff; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 20px;
}

```

```

.topic-header {
  background-color: #e9ecef; padding: 10px 15px; border-bottom: 1px solid #ccc;
  display: flex; justify-content: space-between; align-items: center; font-weight: bold;
}
.topic-stats { font-weight: normal; font-size: 0.9em; }

/* Таблиця */
table { width: 100%; border-collapse: collapse; }
th, td { padding: 10px 15px; text-align: left; border-bottom: 1px solid #eee; }
th { background-color: #f8f9fa; color: #333; font-weight: bold; }
tr:last-child td { border-bottom: none; }

.score-points { font-weight: bold; color: #28a745; }
.score-percent { color: #555; font-size: 0.9em; }

/* Прогноз */
.prediction-box {
  margin-top: 20px; padding: 15px;
  background-color: #fff3cd; border: 1px solid #ffeeba;
  border-radius: 5px; color: #856404;
}
.prediction-box h3 { margin-top: 0; margin-bottom: 10px; font-size: 1.2em; }
.topic-prediction { padding: 10px 15px; background-color: #f0f8ff; border-top: 1px solid #ccc; font-size: 0.9em; color: #0056b3; }

.section-title { margin-top: 0; margin-bottom: 15px; color: #333; font-size: 1.2em; font-weight: bold; }
.no-data { text-align: center; padding: 30px; background: #fff; border: 1px solid #ccc; border-radius: 5px; color: #777; }
</style>
</head>
<body>
<header>
<div>
<h1>Аналіз Успішності</h1>
<span class="class-info">Вітаємо, <?php echo htmlspecialchars($_SESSION['name']);
?> (Клас: <?php echo htmlspecialchars($student_class); ?>)</span>
</div>
<nav>
<a href="student_dashboard.php">Мої Тести</a>
<a href="student_grades.php" class="active">Аналітика</a>
<a href="logout.php">Вийти</a>
</nav>
</header>

<main>
<div class="filter-section">
<label for="subject_select"><strong>Оберіть предмет:</strong></label><br><br>
<form action="student_grades.php" method="GET">
<select name="subject_id" id="subject_select" onchange="this.form.submit()">
<option value="0">-- Всі предмети (Загальна статистика) --</option>
<?php

```

```

        $subjects = $db->query("SELECT * FROM subjects ORDER BY name");
        while ($subject = $subjects->fetch_assoc()) {
            $selected = ($subject['id'] == $selected_subject_id) ? 'selected' : "";
            echo "<option value='{$subject['id']}' $selected>" .
htmlspecialchars($subject['name']) . "</option>";
        }
    ?>
</select>
</form>
</div>

<?php if ($selected_subject_id > 0): ?>

<?php
// логіка для конкретного предмету
// Отримання даних
$sql = "SELECT
        t.topic_name,
        q.id AS quiz_id,
        q.title AS quiz_title,
        qr.score AS percent_score,
        qr.date_taken,
        (SELECT SUM(p.points) FROM questions p WHERE p.quiz_id = q.id) AS
max_points
FROM quiz_results qr
JOIN quizzes q ON qr.quiz_id = q.id
JOIN topics t ON q.topic_id = t.id
WHERE qr.student_id = ? AND t.subject_id = ?
ORDER BY t.topic_name, qr.date_taken DESC";

$stmt = $db->prepare($sql);
$stmt->bind_param("ii", $student_id, $selected_subject_id);
$stmt->execute();
$result = $stmt->get_result();

// Обробка даних
$topics_data = [];
$grand_total_earned = 0;
$grand_total_possible = 0;
$grand_tests_count = 0;
$all_percents_for_general_prediction = [];

while ($row = $result->fetch_assoc()) {
    $topic = $row['topic_name'];
    $max = $row['max_points'] ? $row['max_points'] : 0;
    $earned = ($max > 0) ? round(($row['percent_score'] / 100) * $max, 1) : 0;

    $grand_total_earned += $earned;
    $grand_total_possible += $max;
    $grand_tests_count++;
    $all_percents_for_general_prediction[] = $row['percent_score'];
}

```

```

if (!isset($topics_data[$topic])) {
    $topics_data[$topic] = [
        'total_earned' => 0,
        'total_possible' => 0,
        'tests' => [],
        'history' => []
    ];
}
$topics_data[$topic]['total_earned'] += $earned;
$topics_data[$topic]['total_possible'] += $max;
$topics_data[$topic]['history'][] = $row['percent_score'];
$topics_data[$topic]['tests'][] = [
    'title' => $row['quiz_title'],
    'date' => $row['date_taken'],
    'earned' => $earned,
    'max' => $max,
    'percent' => $row['percent_score']
];
}
$stmt->close();
?>

<?php if ($grand_tests_count > 0): ?>
<?php
    $grand_avg_percent = array_sum($all_percents_for_general_prediction) /
count($all_percents_for_general_prediction);
?>
<h3 class="section-title">Статистика з обраного предмету</h3>
<div class="summary-card">
    <div class="summary-item">
        <h3><?php echo $grand_tests_count; ?></h3>
        <p>Пройдено тестів</p>
    </div>
    <div class="summary-item">
        <h3><?php echo $grand_total_earned; ?> <span style="font-size:0.5em;
color:#777;">/ <?php echo $grand_total_possible; ?></span></h3>
        <p>Загальний бал</p>
    </div>
    <div class="summary-item">
        <h3><?php echo round($grand_avg_percent, 1); ?>%</h3>
        <p>Середня успішність</p>
    </div>
</div>

<?php foreach ($topics_data as $topic_name => $data): ?>
<?php
    $topic_percent = ($data['total_possible'] > 0) ? ($data['total_earned'] /
$data['total_possible']) * 100 : 0;
    $topic_history_asc = array_reverse($data['history']);
    $topic_prediction = predictNextScore($topic_history_asc);
?>
<div class="topic-block">

```

```

<div class="topic-header">
  <span><?php echo htmlspecialchars($topic_name); ?></span>
  <span class="topic-stats">
    Всього: <?php echo $data['total_earned']; ?> / <?php echo
    $data['total_possible']; ?> б.
    (<?php echo round($topic_percent, 1); ?>% )
  </span>
</div>
<table>
  <thead>
    <tr>
      <th width="40%">Тест</th>
      <th width="20%">Дата</th>
      <th width="20%">Бали</th>
      <th width="20%">Результат</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($data['tests'] as $test): ?>
      <tr>
        <td><?php echo htmlspecialchars($test['title']); ?></td>
        <td><?php echo date("d.m.Y H:i", strtotime($test['date'])); ?></td>
        <td><span class="score-points"><?php echo $test['earned']; ?> / <?php
        echo $test['max']; ?></span></td>
        <td><span class="score-percent"><?php echo $test['percent'];
        ?>%</span></td>
      </tr>
    <?php endforeach; ?>
  </tbody>
</table>
<?php if ($topic_prediction !== null): ?>
  <div class="topic-prediction">
    Прогноз на наступний тест з теми "<?php echo
    htmlspecialchars($topic_name); ?>": <strong><?php echo $topic_prediction; ?>%</strong>
  </div>
<?php else: ?>
  <div class="topic-prediction" style="color: #777; font-style: italic;">
    Недостатньо даних для прогнозу по цій темі (потрібно мінімум 3 тести).
  </div>
<?php endif; ?>
</div>
<?php endforeach; ?>

<div class="prediction-box">
  <h3>Загальний прогноз по предмету</h3>
  <?php
  $general_history_asc = array_reverse($all_percents_for_general_prediction);
  $general_prediction = predictNextScore($general_history_asc);
  ?>
  <?php if ($general_prediction !== null): ?>

```

```

        <p>На основі динаміки всіх ваших тестів (<?php echo $grand_tests_count; ?>
шт.), очікувана оцінка за <strong>наступний тест з цього предмету</strong>:
<strong><?php echo $general_prediction; ?>%</strong></p>

```

```

        <?php else: ?>
        <p>Пройдіть ще кілька тестів, щоб система могла спрогнозувати ваш
результат (потрібно мінімум 3).</p>

```

```

        <?php endif; ?>
    </div>

```

```

    <?php else: ?>
        <div class="no-data">Ви ще не проходили тестів з цього предмету.</div>
    <?php endif; ?>

```

```

<?php else: ?>

```

```

<?php
// статистика по усіх предметах

```

```

$total_all_earned = 0;
$total_all_possible = 0;
$total_all_tests = 0;
$all_scores_sum = 0;

```

```

// Отримуємо всі результати студента незалежно від предмету

```

```

$sql_all = "SELECT
            qr.score AS percent_score,
            (SELECT SUM(p.points) FROM questions p WHERE p.quiz_id = q.id) AS
max_points
            FROM quiz_results qr
            JOIN quizzes q ON qr.quiz_id = q.id
            WHERE qr.student_id = ?";

```

```

$stmt_all = $db->prepare($sql_all);
$stmt_all->bind_param("i", $student_id);
$stmt_all->execute();
$res_all = $stmt_all->get_result();

```

```

while($row = $res_all->fetch_assoc()) {
    $max = $row['max_points'] ? $row['max_points'] : 0;
    $earned = ($max > 0) ? round(($row['percent_score'] / 100) * $max, 1) : 0;

```

```

        $total_all_earned += $earned;
        $total_all_possible += $max;
        $total_all_tests++;
        $all_scores_sum += $row['percent_score'];
    }

```

```

    $stmt_all->close();
    ?>

```

```

<h3 class="section-title">Загальна статистика (всі предмети)</h3>

```

```

<?php if ($total_all_tests > 0): ?>

```

```

<?php
    $avg_all = $all_scores_sum / $total_all_tests;
?>
<div class="summary-card">
    <div class="summary-item">
        <h3><?php echo $total_all_tests; ?></h3>
        <p>Всього тестів пройдено</p>
    </div>
    <div class="summary-item">
        <h3><?php echo $total_all_earned; ?> <span style="font-size:0.5em;
color:#777;">/ <?php echo $total_all_possible; ?></span></h3>
        <p>Загальний бал (сумарно)</p>
    </div>
    <div class="summary-item">
        <h3><?php echo round($avg_all, 1); ?>%</h3>
        <p>Загальна середня успішність</p>
    </div>
</div>

<div style="text-align: center; margin-top: 40px; color: #555;">
    <p>Оберіть конкретний предмет зі списку вище, щоб побачити деталізацію по
темах та прогнозування.</p>
</div>

<?php else: ?>
    <div class="no-data">Ви ще не пройшли жодного тесту в системі.</div>
<?php endif; ?>

<?php endif; ?>
</main>
</body>
</html>

```

Код файлу «submit_test.php»:

```

<?php
include 'config.php';
check_auth(2);
header('Content-Type: application/json');

if ($_SERVER["REQUEST_METHOD"] != "POST") {
    echo json_encode(['success' => false, 'message' => 'Invalid request']);
    exit;
}

$student_id = $_SESSION['user_id'];
$quiz_id = (int)$_POST['quiz_id'];
$student_answers_post = isset($_POST['answers']) ? $_POST['answers'] : [];
$student_text_answers = isset($_POST['text_answers']) ? $_POST['text_answers'] : [];

```

```

// Збираємо інфо про питання та макс бали
$questions_info = [];
$total_possible_score = 0;
$has_text_questions = false;

$q_stmt = $db->prepare("SELECT id, points, type FROM questions WHERE quiz_id = ?");
$q_stmt->bind_param("i", $quiz_id);
$q_stmt->execute();
$q_res = $q_stmt->get_result();
while ($q = $q_res->fetch_assoc()) {
    $questions_info[$q['id']] = [
        'points' => $q['points'],
        'type' => $q['type'],
        'correct_ids' => [],
    ];
    $total_possible_score += $q['points'];
    if ($q['type'] == 'text') $has_text_questions = true;
}
$q_stmt->close();

// Отримуємо правильні відповіді для авто-перевірки
foreach ($questions_info as $qid => $info) {
    if ($info['type'] != 'text') {
        $a_stmt = $db->prepare("SELECT id FROM answers WHERE question_id = ? AND
is_correct = 1");
        $a_stmt->bind_param("i", $qid);
        $a_stmt->execute();
        $a_res = $a_stmt->get_result();
        while ($a = $a_res->fetch_assoc()) {
            $questions_info[$qid]['correct_ids'][] = $a['id'];
        }
        $a_stmt->close();
    }
}

// Збереження відповідей та підрахунок
$earned_score = 0;
$db->begin_transaction();

try {
    $status = $has_text_questions ? 'pending' : 'completed';
    $result_stmt = $db->prepare("INSERT INTO quiz_results (student_id, quiz_id, score, status,
date_taken) VALUES (?, ?, 0, ?, NOW())");
    $result_stmt->bind_param("iis", $student_id, $quiz_id, $status);
    $result_stmt->execute();
    $result_id = $db->insert_id;
    $result_stmt->close();

    $save_answ_stmt = $db->prepare("INSERT INTO user_answers (result_id, question_id,
chosen_answer_id, text_answer, points_awarded) VALUES (?, ?, ?, ?, ?)");
}

// А. Обробка тестових питань

```

```

foreach ($student_answers_post as $q_id => $user_ans) {
    if (!isset($questions_info[$q_id])) continue;
    $q_data = $questions_info[$q_id];
    $selected_ids = is_array($user_ans) ? $user_ans : [$user_ans];
    $correct_ids = $q_data['correct_ids'];
    $points_for_q = $q_data['points'];

    $q_score = 0;
    if ($q_data['type'] == 'single') {
        if (count($selected_ids) == 1 && in_array($selected_ids[0], $correct_ids)) $q_score =
$points_for_q;
    } elseif ($q_data['type'] == 'multiple') {
        $total_correct = count($correct_ids);
        if ($total_correct > 0) {
            $sval = $points_for_q / $total_correct;
            foreach ($selected_ids as $sid) {
                if (in_array($sid, $correct_ids)) $q_score += $sval;
                else $q_score -= $sval; // Штраф за неправильну
            }
        }
    }
    if ($q_score < 0) $q_score = 0;
    $earned_score += $q_score;

    // Зберігаємо відповіді. Бали записуємо тільки в перший рядок для цього питання
    $is_first = true;
    foreach ($selected_ids as $sid) {
        $txt = null;
        $pts = $is_first ? $q_score : 0;
        $save_anw_stmt->bind_param("iisd", $result_id, $q_id, $sid, $txt, $pts);
        $save_anw_stmt->execute();
        $is_first = false;
    }
}

// Б. Обробка текстових питань
foreach ($student_text_answers as $q_id => $text) {
    $text = trim($text);
    $sid = null;
    $pts = 0; // 0 балів до перевірки вчителем
    $save_anw_stmt->bind_param("iisd", $result_id, $q_id, $sid, $text, $pts);
    $save_anw_stmt->execute();
}
$save_anw_stmt->close();

// Оновлюємо результат в БД
$percentage_score = ($total_possible_score > 0) ? ($earned_score / $total_possible_score) *
100 : 0;
$percentage_score = round($percentage_score, 2);
$earned_score = round($earned_score, 2);

$supd = $db->prepare("UPDATE quiz_results SET score = ? WHERE id = ?");

```

```

$upd->bind_param("di", $percentage_score, $result_id);
$upd->execute();
$upd->close();

$db->commit();

$msg = $has_text_questions ? "<span style='color:orange'>Відправлено на перевірку
вчителю.</span>" : "Тест завершено!";

echo json_encode([
    'success' => true,
    'earned_points' => $earned_score,
    'total_points' => $total_possible_score,
    'percentage' => $percentage_score,
    'message' => $msg
]);

} catch (Exception $e) {
    $db->rollback();
    echo json_encode(['success' => false, 'message' => 'Error: ' . $e->getMessage()]);
}
?>

```

Код файлу «take_test.php»:

```

<?php
include 'config.php';
check_auth(2); // Тільки учень

if (!isset($_GET['id'])) { header('Location: student_dashboard.php'); exit; }
$quiz_id = (int)$_GET['id'];
$student_id = $_SESSION['user_id'];

// Перевіряємо, чи тест вже не пройдено
$check_stmt = $db->prepare("SELECT id FROM quiz_results WHERE student_id = ? AND
quiz_id = ?");
$check_stmt->bind_param("ii", $student_id, $quiz_id);
$check_stmt->execute();
if ($check_stmt->get_result()->num_rows > 0) {
    die("Ви вже пройшли цей тест. <a href='student_dashboard.php'>Повернутись</a>");
}
$check_stmt->close();

// Отримуємо інформацію про тест (назва, час)
$quiz_stmt = $db->prepare("SELECT title, time_limit FROM quizzes WHERE id = ?");
$quiz_stmt->bind_param("i", $quiz_id);
$quiz_stmt->execute();
$quiz = $quiz_stmt->get_result()->fetch_assoc();
if (!$quiz) die("Тест не знайдено.");
?>

```

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Тестування</title>
  <style>
    /* Стили ідентичні іншим сторінкам */
    body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4; color: #333;
  }
    header { display: flex; justify-content: space-between; align-items: center; padding-bottom:
10px; border-bottom: 2px solid #ccc; margin-bottom: 20px; }
    .timer-box { font-size: 18px; font-weight: bold; color: #d9534f; background-color: #fff;
padding: 5px 15px; border: 1px solid #ccc; border-radius: 4px; }
    .question-container { width: 70%; margin: 0 auto 50px auto; background-color: #fff; border:
2px solid #000; padding: 20px; border-radius: 10px; box-shadow: 0 0 10px rgba(0,0,0,0.1); }
    .question-card { margin-bottom: 20px; padding-bottom: 20px; border-bottom: 1px solid
#ccc; }
    .question-card:last-child { border-bottom: none; }
    .question-card h3 { margin-top: 0; font-size: 18px; border-bottom: 1px solid #eee; padding-
bottom: 10px; margin-bottom: 15px; }
    .points-badge { font-size: 14px; color: #666; font-weight: normal; float: right; }
    .answers label { display: block; padding: 10px; border-bottom: 1px solid #f0f0f0; cursor:
pointer; }
    .answers label:hover { background-color: #f1f1f1; }
    .submit-btn { display: block; width: 200px; margin: 30px auto; padding: 12px; background-
color: #28a745; color: white; border: 2px solid #000; border-radius: 5px; cursor: pointer; font-
size: 18px; }

    /* Модальне вікно */
    .modal-overlay { display: none; position: fixed; z-index: 1000; left: 0; top: 0; width: 100%;
height: 100%; background-color: rgba(0,0,0,0.5); align-items: center; justify-content: center; }
    .modal-box { background-color: #fff; width: 400px; padding: 25px; border-radius: 8px;
border: 1px solid #ccc; text-align: center; box-shadow: 0 4px 15px rgba(0,0,0,0.2); }
    .modal-title { font-size: 22px; font-weight: bold; margin-bottom: 15px; border-bottom: 1px
solid #eee; padding-bottom: 10px; }
    .result-value { font-weight: bold; color: #28a745; }
    .modal-btn { display: inline-block; margin-top: 20px; padding: 10px 25px; background-
color: #007bff; color: white; text-decoration: none; border-radius: 4px; font-size: 16px; }
  </style>
</head>
<body>
  <header>
    <h1><?php echo htmlspecialchars($quiz['title']); ?></h1>
    <div class="timer-box">Час: <span id="time-display">00:00</span></div>
  </header>

  <form id="testForm" class="question-container">
    <input type="hidden" name="quiz_id" value="<?php echo $quiz_id; ?>">

    <?php
    $q_stmt = $db->prepare("SELECT id, question_text, points, type FROM questions
WHERE quiz_id = ?");

```

```

$q_stmt->bind_param("i", $quiz_id);
$q_stmt->execute();
$questions = $q_stmt->get_result();

$q_num = 1;
while ($q = $questions->fetch_assoc()) {
    // Підказка типу питання для студента
    if ($q['type'] == 'multiple') $typeText = '(Оберіть всі правильні відповіді)';
    elseif ($q['type'] == 'text') $typeText = '(Введіть відповідь вручну)';
    else $typeText = '(Оберіть одну відповідь)';

    echo "<div class='question-card'>";
    echo "<h3>" . $q_num . ". " . htmlspecialchars($q['question_text']) .
        " <br><small style='color:#007bff;'>$typeText</small>" .
        " <span class='points-badge'>Балів: ".$q['points'].</span></h3>";
    echo "<div class='answers'>";

    if ($q['type'] == 'text') {
        // Поле для тексту
        echo "<textarea name='text_answers[" . $q['id'] . "]" rows='4' style='width:100%;
border-radius:4px; border:1px solid #ccc; padding:10px;' placeholder='Введіть вашу
відповідь...'></textarea>";
    } else {
        // Варіанти відповідей
        $a_stmt = $db->prepare("SELECT id, answer_text FROM answers WHERE
question_id = ?");
        $a_stmt->bind_param("i", $q['id']);
        $a_stmt->execute();
        $answers = $a_stmt->get_result();

        $ans_list = [];
        while ($a = $answers->fetch_assoc()) { $ans_list[] = $a; }
        $a_stmt->close();
        shuffle($ans_list);

        foreach ($ans_list as $a) {
            $radio_id = "ans_" . $a['id'];
            echo "<label for='$radio_id'>";
            if ($q['type'] == 'multiple') {
                echo "<input type='checkbox' name='answers[" . $q['id'] . "]" value='' . $a['id'] .
'" id='$radio_id'>";
            } else {
                echo "<input type='radio' name='answers[" . $q['id'] . "]" value='' . $a['id'] . '"
id='$radio_id'>";
            }
            echo " " . htmlspecialchars($a['answer_text']);
            echo "</label>";
        }
    }
    echo "</div></div>";
    $q_num++;
}

```

```

?>
<button type="submit" class="submit-btn">Завершити тест</button>
</form>

<div id="resultModal" class="modal-overlay">
  <div class="modal-box">
    <div class="modal-title">Результат тесту</div>
    <div class="result-text" id="modalMessage"></div>
    <div class="result-text">Набрано балів: <span class="result-value"
id="modalPoints">0/0</span></div>
    <div class="result-text">Оцінка: <span class="result-value"
id="modalPercent">0%</span></div>
    <a href="student_dashboard.php" class="modal-btn">На головну</a>
  </div>
</div>

<script>
  // Таймер
  let timeRemaining = <?php echo $quiz['time_limit'] * 60; ?>;
  const display = document.getElementById('time-display');
  const form = document.getElementById('testForm');
  const modal = document.getElementById('resultModal');

  function updateTimer() {
    let minutes = Math.floor(timeRemaining / 60);
    let seconds = timeRemaining % 60;
    display.textContent = `${minutes}:${seconds < 10 ? '0' : ''}${seconds}`;
    if (timeRemaining <= 0) { clearInterval(timerInterval); finishTest(); }
    timeRemaining--;
  }
  const timerInterval = setInterval(updateTimer, 1000);
  updateTimer();

  form.addEventListener('submit', function(e) {
    e.preventDefault();
    if(confirm('Завершити виконання тесту?')) finishTest();
  });

  async function finishTest() {
    clearInterval(timerInterval);
    const formData = new FormData(form);
    try {
      const response = await fetch('submit_test.php', { method: 'POST', body: formData });
      const data = await response.json();
      if (data.success) {
        document.getElementById('modalPercent').textContent = data.percentage + '%';
        document.getElementById('modalPoints').textContent = data.earned_points + ' з ' +
data.total_points;
        document.getElementById('modalMessage').innerHTML = data.message;
        modal.style.display = 'flex';
      } else { alert('Помилка: ' + data.message); }
    } catch (error) { console.error(error); alert('Помилка з\'єднання.'); }
  }
</script>

```

```

    }
    window.onbeforeunload = function() { if (modal.style.display !== 'flex') return "Тест не
завершено!"; };
</script>
</body>
</html>

```

Код файлу «teacher_dashboard.php»:

```

<?php
include 'config.php';
check_auth(1); // Тільки для вчителів (роль 1)

// Головна сторінка вчителя
header('Location: teacher_quizzes.php');
exit;
?>

```

Код файлу «teacher_grades.php»:

```

<?php
include 'config.php';
include 'prediction.php'; // Підключення прогнозу оцінок
check_auth(1);
$teacher_id = $_SESSION['user_id'];
$subject_id = $_SESSION['teacher_subject_id'];

// Отримка назви предмету
$subject_name = "Не призначено";
if ($subject_id) {
    $sub_stmt = $db->prepare("SELECT name FROM subjects WHERE id = ?");
    $sub_stmt->bind_param("i", $subject_id);
    $sub_stmt->execute();
    $sub_res = $sub_stmt->get_result();
    if ($row = $sub_res->fetch_assoc()) { $subject_name = $row['name']; }
    $sub_stmt->close();
}

$selected_class = isset($_GET['class_name']) ? $_GET['class_name'] : "";
$selected_student_id = isset($_GET['student_id']) ? (int)$_GET['student_id'] : 0;

// Отримка імені обраного студента
$selected_student_name = "";
if ($selected_student_id) {
    $s_name_q = $db->query("SELECT name FROM users WHERE id = $selected_student_id");

```

```

    if ($s_row = $s_name_q->fetch_assoc()) {
        $selected_student_name = $s_row['name'];
    }
}
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Журнал Оцінок</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4; color: #333;
    }

    /* шапка */
    header {
        display: flex;
        justify-content: space-between;
        align-items: center;
        padding-bottom: 10px;
        border-bottom: 2px solid #ccc;
        margin-bottom: 20px;
        background-color: #fff;
        padding: 15px 20px;
        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0,0,0,0.05);
    }
    header h1 { margin: 0; font-size: 24px; }
    .user-info { color: #777; font-size: 0.9em; margin-top: 5px; display: block;}

    /* навігація */
    nav { display: flex; gap: 10px; }
    nav a {
        text-decoration: none; color: white; background-color: #6c757d;
        padding: 8px 16px; border-radius: 4px; font-weight: bold; font-size: 14px;
        transition: background-color 0.2s;
    }
    nav a:hover { background-color: #5a6268; }
    nav a.active { background-color: #495057; cursor: default; box-shadow: inset 0 2px 4px
    rgba(0,0,0,0.2); }
    nav a[href="logout.php"] { background-color: #dc3545; }
    nav a[href="logout.php"]:hover { background-color: #c82333; }

    /* фільтр */
    .filters { background: #fff; padding: 20px; border-radius: 5px; border: 1px solid #ccc;
    display: flex; gap: 20px; align-items: flex-end; margin-bottom: 20px; }
    .filter-group { display: flex; flex-direction: column; }
    label { font-weight: bold; margin-bottom: 5px; font-size: 14px; }
    select { padding: 8px; min-width: 200px; border: 1px solid #ccc; border-radius: 4px; }
    button.btn-reset { padding: 8px 15px; background-color: #6c757d; color: white; border:
    none; border-radius: 4px; cursor: pointer; }
    button.btn-reset:hover { background-color: #5a6268; }

```

```

/* стилі для статистики */
.summary-card {
  background-color: #fff; border: 1px solid #ccc; padding: 20px;
  border-radius: 5px; margin-bottom: 20px; display: flex;
  justify-content: space-around; text-align: center;
}
.summary-item h3 { margin: 5px 0; font-size: 2em; color: #007bff; }
.summary-item p { margin: 0; color: #555; font-size: 1em; }

.topic-block {
  background: #fff; border: 1px solid #ccc; border-radius: 5px; margin-bottom: 20px;
}
.topic-header {
  background-color: #e9ecef; padding: 10px 15px; border-bottom: 1px solid #ccc;
  display: flex; justify-content: space-between; align-items: center; font-weight: bold;
}
.topic-stats { font-weight: normal; font-size: 0.9em; }

.score-points { font-weight: bold; color: #28a745; }
.score-percent { color: #555; font-size: 0.9em; }

.prediction-box {
  margin-top: 20px; padding: 15px;
  background-color: #fff3cd; border: 1px solid #ffe5a3;
  border-radius: 5px; color: #856404;
}
.prediction-box h3 { margin-top: 0; margin-bottom: 10px; font-size: 1.2em; }
.topic-prediction { padding: 10px 15px; background-color: #f0f8ff; border-top: 1px solid
#ccc; font-size: 0.9em; color: #0056b3; }

/* таблиця */
.simple-table { width: 100%; border-collapse: collapse; background: #fff; box-shadow: 0
2px 5px rgba(0,0,0,0.1); border-radius: 5px; overflow: hidden; }
.simple-table th, .simple-table td { padding: 12px 15px; border-bottom: 1px solid #eee; text-
align: left; }
.simple-table th { background-color: #f8f9fa; color: #333; font-weight: bold; }

.no-data { padding: 30px; text-align: center; color: #777; background: #fff; border-radius:
5px; border: 1px solid #ccc; }
.student-title { font-size: 1.5em; margin-bottom: 15px; color: #333; border-bottom: 2px
solid #007bff; display: inline-block; padding-bottom: 5px; }
</style>
</head>
<body>
<header>
<div>
<h1>Журнал Оцінок</h1>
<span class="user-info">Вітаємо, <?php echo htmlspecialchars($_SESSION['name']);
?> (Викладач з предмету: <?php echo htmlspecialchars($subject_name); ?>)</span>
</div>
<nav>

```

```

    <a href="teacher_quizzes.php">Мої Тести</a>
    <a href="teacher_grades.php" class="active">Оцінки Учні</a>
    <a href="logout.php">Вийти</a>
  </nav>
</header>

<main>
  <form class="filters" method="GET">
    <div class="filter-group">
      <label for="class_name">Оберіть Клас:</label>
      <select name="class_name" id="class_name" onchange="this.form.submit()">
        <option value="">-- Всі класи --</option>
        <?php
          $class_sql = "SELECT DISTINCT class_name FROM quizzes WHERE teacher_id
= ? ORDER BY class_name";
          $c_stmt = $db->prepare($class_sql);
          $c_stmt->bind_param("i", $teacher_id);
          $c_stmt->execute();
          $c_res = $c_stmt->get_result();
          while($row = $c_res->fetch_assoc()){
            $sel = ($selected_class == $row['class_name']) ? 'selected' : "";
            echo "<option value='".$row['class_name']."'
$sel>".$row['class_name']."</option>";
          }
        ?>
      </select>
    </div>

    <?php if($selected_class): ?>
    <div class="filter-group">
      <label for="student_id">Оберіть Учня:</label>
      <select name="student_id" id="student_id" onchange="this.form.submit()">
        <option value="0">-- Всі учні класу (Загальний список) --</option>
        <?php
          $stud_sql = "SELECT id, name FROM users WHERE role = 2 AND class_name =
? ORDER BY name";
          $s_stmt = $db->prepare($stud_sql);
          $s_stmt->bind_param("s", $selected_class);
          $s_stmt->execute();
          $s_res = $s_stmt->get_result();
          while($row = $s_res->fetch_assoc()){
            $sel = ($selected_student_id == $row['id']) ? 'selected' : "";
            echo "<option value='".$row['id']."' $sel>".$row['name']."</option>";
          }
        ?>
      </select>
    </div>
    <?php endif; ?>

    <div class="filter-group">
      <label>&nbsp;</label>

```

```

        <a href="teacher_grades.php"><button type="button" class="btn-reset">Скинути
фільтри</button></a>
    </div>
</form>

<?php if($selected_class): ?>

    <?php if($selected_student_id > 0): ?>
        <h2 class="student-title">Статистика учня: <?php echo
htmlspecialchars($selected_student_name); ?></h2>

        <?php
        $sql = "SELECT
            t.topic_name,
            q.id AS quiz_id,
            q.title AS quiz_title,
            qr.score AS percent_score,
            qr.date_taken,
            (SELECT SUM(p.points) FROM questions p WHERE p.quiz_id = q.id) AS
max_points
            FROM quiz_results qr
            JOIN quizzes q ON qr.quiz_id = q.id
            JOIN topics t ON q.topic_id = t.id
            WHERE qr.student_id = ? AND q.teacher_id = ?
            ORDER BY t.topic_name, qr.date_taken DESC";

        $stmt = $db->prepare($sql);
        $stmt->bind_param("ii", $selected_student_id, $teacher_id);
        $stmt->execute();
        $result = $stmt->get_result();

        $topics_data = [];
        $grand_total_earned = 0;
        $grand_total_possible = 0;
        $grand_tests_count = 0;
        $all_percentages_for_prediction = [];

        while ($row = $result->fetch_assoc()) {
            $topic = $row['topic_name'];
            $max = $row['max_points'] ? $row['max_points'] : 0;
            $earned = ($max > 0) ? round(($row['percent_score'] / 100) * $max, 1) : 0;

            $grand_total_earned += $earned;
            $grand_total_possible += $max;
            $grand_tests_count++;
            $all_percentages_for_prediction[] = $row['percent_score'];

            if (!isset($topics_data[$topic])) {
                $topics_data[$topic] = [
                    'total_earned' => 0, 'total_possible' => 0, 'tests' => [], 'history' => []
                ];
            }
        }
    </?php if($selected_student_id > 0): ?>
</?php if($selected_class): ?>

```

```

    $topics_data[$topic]['total_earned'] += $earned;
    $topics_data[$topic]['total_possible'] += $max;
    $topics_data[$topic]['history'][] = $row['percent_score'];
    $topics_data[$topic]['tests'][] = [
        'title' => $row['quiz_title'],
        'date' => $row['date_taken'],
        'earned' => $earned,
        'max' => $max,
        'percent' => $row['percent_score']
    ];
}
$stmt->close();
?>

<?php if ($grand_tests_count > 0): ?>
    <?php $grand_avg_percent = array_sum($all_percents_for_prediction) /
count($all_percents_for_prediction); ?>
    <div class="summary-card">
        <div class="summary-item">
            <h3><?php echo $grand_tests_count; ?></h3>
            <p>Пройдено тестів</p>
        </div>
        <div class="summary-item">
            <h3><?php echo $grand_total_earned; ?> <span style="font-size:0.5em;
color:#777;">/ <?php echo $grand_total_possible; ?></span></h3>
            <p>Загальний бал</p>
        </div>
        <div class="summary-item">
            <h3><?php echo round($grand_avg_percent, 1); ?>%</h3>
            <p>Середня успішність</p>
        </div>
    </div>

    <?php foreach ($topics_data as $topic_name => $data): ?>
        <?php
            $topic_percent = ($data['total_possible'] > 0) ? ($data['total_earned'] /
$data['total_possible']) * 100 : 0;
            // Прогноз балів за конкретною темою
            $topic_history_asc = array_reverse($data['history']);
            $topic_prediction = predictNextScore($topic_history_asc);
        ?>
        <div class="topic-block">
            <div class="topic-header">
                <span><?php echo htmlspecialchars($topic_name); ?></span>
                <span class="topic-stats">
                    Всього: <?php echo $data['total_earned']; ?> / <?php echo
$data['total_possible']; ?> б.
                    (<?php echo round($topic_percent, 1); ?>%)
                </span>
            </div>
            <table class="simple-table">
                <thead>

```

```

        <tr>
            <th width="40%">Тест</th>
            <th width="20%">Дата</th>
            <th width="20%">Бали</th>
            <th width="20%">Результат</th>
        </tr>
    </thead>
    <tbody>
        <?php foreach ($data['tests'] as $test): ?>
            <tr>
                <td><?php echo htmlspecialchars($test['title']); ?></td>
                <td><?php echo date("d.m.Y H:i", strtotime($test['date'])); ?></td>
                <td><span class="score-points"><?php echo $test['earned']; ?> /
<?php echo $test['max']; ?></span></td>
                <td><span class="score-percent"><?php echo $test['percent'];
?>%</span></td>
            </tr>
        <?php endforeach; ?>
    </tbody>
</table>
<?php if ($topic_prediction !== null): ?>
    <div class="topic-prediction">
        Прогноз на наступний тест з теми "<?php echo
htmlspecialchars($topic_name); ?>": <strong><?php echo $topic_prediction; ?>%</strong>
    </div>
<?php endif; ?>
</div>
<?php endforeach; ?>

<div class="prediction-box">
    <h3>Прогноз успішності учня</h3>
    <?php
$general_history_asc = array_reverse($all_percents_for_prediction);
$general_prediction = predictNextScore($general_history_asc);
?>
    <?php if ($general_prediction !== null): ?>
        <p>Очікувана оцінка за наступний тест з предмету "<?php echo
htmlspecialchars($subject_name); ?>": <strong><?php echo $general_prediction;
?>%</strong></p>
    <?php else: ?>
        <p>Недостатньо даних для прогнозування (потрібно мінімум 3
тести).</p>
    <?php endif; ?>
</div>

<?php else: ?>
    <div class="no-data">Цей учень ще не проходив тестів з вашого
предмету.</div>
<?php endif; ?>

<?php else: ?>
    <table class="simple-table">

```

```

<thead>
  <tr>
    <th>Учень</th>
    <th>Тест</th>
    <th>Тема</th>
    <th>Дата задачі</th>
    <th>Бали</th> <th>Оцінка (%)</th>
  </tr>
</thead>
<tbody>
  <?php
    $query = "SELECT
      u.name AS student_name,
      q.title AS quiz_title,
      t.topic_name,
      qr.score,
      qr.date_taken,
      (SELECT SUM(p.points) FROM questions p WHERE p.quiz_id = q.id)
AS max_points
      FROM quizzes q
      JOIN users u ON u.class_name = q.class_name AND u.role = 2
      LEFT JOIN quiz_results qr ON qr.quiz_id = q.id AND qr.student_id = u.id
      JOIN topics t ON q.topic_id = t.id
      WHERE q.teacher_id = ? AND q.class_name = ?
      ORDER BY u.name, q.title";

    $res_stmt = $db->prepare($query);
    $res_stmt->bind_param("is", $teacher_id, $selected_class);
    $res_stmt->execute();
    $results = $res_stmt->get_result();

    if($results->num_rows > 0) {
      while($row = $results->fetch_assoc()) {
        // Розрахунок балів
        $max = $row['max_points'] ? $row['max_points'] : 0;
        $earned = ($row['score'] !== null) ? round(($row['score'] / 100) * $max, 1) :
'-';

        echo "<tr>";
        echo "<td>" . htmlspecialchars($row['student_name']) . "</td>";
        echo "<td>" . htmlspecialchars($row['quiz_title']) . "</td>";
        echo "<td>" . htmlspecialchars($row['topic_name']) . "</td>";
        echo "<td>" . ($row['date_taken'] ? date("d.m.Y H:i",
strtotime($row['date_taken'])) : '<span style="color:gray">Не здавав</span>') . "</td>";
        echo "<td>" . (($row['score'] !== null) ? "$earned / $max" : '-') . "</td>"; //
Бали
        echo "<td>" . ($row['score'] !== null ? "<strong>".$row['score'].</strong>"
: '-') . "</td>";
        echo "</tr>";
      }
    } else {
      echo "<tr><td colspan='6' class='no-data'>Даних не знайдено.</td></tr>";
    }
  }

```

```

        }
        ?>
    </tbody>
</table>
<?php endif; ?>

<?php else: ?>
    <div class="no-data">Будь ласка, оберіть клас, щоб переглянути журнал.</div>
<?php endif; ?>
</main>
</body>
</html>

```

Код файлу «teacher_quiz_students.php»:

```

<?php
include 'config.php';
check_auth(1);
$teacher_id = $_SESSION['user_id'];
$subject_id = $_SESSION['teacher_subject_id'];

// Отримка назви предмету
$subject_name = "";
if ($subject_id) {
    $sub = $db->query("SELECT name FROM subjects WHERE id = $subject_id")-
>fetch_assoc();
    $subject_name = $sub['name'];
}

if (!isset($_GET['quiz_id'])) {
    header('Location: teacher_quizzes.php');
    exit;
}
$quiz_id = (int)$_GET['quiz_id'];

$check = $db->prepare("SELECT title, class_name FROM quizzes WHERE id = ? AND
teacher_id = ?");
$check->bind_param("ii", $quiz_id, $teacher_id);
$check->execute();
$quiz_info = $check->get_result()->fetch_assoc();
if (!$quiz_info) die("Тест не знайдено.");
$check->close();
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Результати тесту</title>

```

```

<style>
  body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4; color: #333;
}

/* шапка */
header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding-bottom: 10px;
  border-bottom: 2px solid #ccc;
  margin-bottom: 20px;
  background-color: #fff;
  padding: 15px 20px;
  border-radius: 5px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.05);
}
header h1 { margin: 0; font-size: 24px; }
.user-info { color: #777; font-size: 0.9em; margin-top: 5px; display: block;}

/* навігація */
nav { display: flex; gap: 10px; }
nav a {
  text-decoration: none; color: white; background-color: #6c757d;
  padding: 8px 16px; border-radius: 4px; font-weight: bold; font-size: 14px;
  transition: background-color 0.2s;
}
nav a:hover { background-color: #5a6268; }
nav a.active { background-color: #495057; cursor: default; box-shadow: inset 0 2px 4px
rgba(0,0,0,0.2); }
nav a[href="logout.php"] { background-color: #dc3545; }
nav a[href="logout.php"]:hover { background-color: #c82333; }

/* Підзаголовок */
.sub-header {
  margin-bottom: 20px;
  background: #fff; padding: 15px; border-radius: 5px; border: 1px solid #ccc;
  display: flex; justify-content: space-between; align-items: center;
}
.quiz-details { font-size: 1.1em; }
.back-link { color: #007bff; text-decoration: none; font-weight: bold; }

/* Таблиця */
table { width: 100%; border-collapse: collapse; background: #fff; box-shadow: 0 2px 5px
rgba(0,0,0,0.1); border-radius: 5px; overflow: hidden; }
th, td { padding: 12px 15px; border-bottom: 1px solid #eee; text-align: left; }
th { background-color: #f8f9fa; color: #333; font-weight: bold; }
.btn-view { background-color: #007bff; color: white; padding: 6px 12px; text-decoration:
none; border-radius: 4px; font-size: 13px; }
.btn-view:hover { background-color: #0056b3; }
.score-good { color: green; font-weight: bold; }
.score-bad { color: red; }

```

```

</style>
</head>
<body>
  <header>
    <div>
      <h1>Мої Тести</h1>
      <span class="user-info">Вітаємо, <?php echo htmlspecialchars($_SESSION['name']);
?> (Викладач з предмету: <?php echo htmlspecialchars($subject_name); ?>)</span>
    </div>
    <nav>
      <a href="teacher_quizzes.php" class="active">Мої Тести</a>
      <a href="teacher_grades.php">Оцінки Учнів</a>
      <a href="logout.php">Вийти</a>
    </nav>
  </header>

  <div class="sub-header">
    <div class="quiz-details">
      Результати тесту: <strong><?php echo htmlspecialchars($quiz_info['title']);
?></strong>
      (Клас: <?php echo htmlspecialchars($quiz_info['class_name']); ?>)
    </div>
    <a href="teacher_quizzes.php" class="back-link">← Назад до списку</a>
  </div>

  <main>
    <table>
      <thead>
        <tr>
          <th>Учень</th>
          <th>Дата проходження</th>
          <th>Результат (%)</th>
          <th>Дії</th>
        </tr>
      </thead>
      <tbody>
        <?php
        $sql = "SELECT u.id as student_id, u.name, qr.score, qr.date_taken
        FROM quiz_results qr
        JOIN users u ON qr.student_id = u.id
        WHERE qr.quiz_id = ? ORDER BY u.name";

        $stmt = $db->prepare($sql);
        $stmt->bind_param("i", $quiz_id);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
          while ($row = $result->fetch_assoc()) {
            $score_class = ($row['score'] >= 50) ? 'score-good' : 'score-bad';
            echo "<tr>";
            echo "<td>" . htmlspecialchars($row['name']) . "</td>";

```

```

        echo "<td>" . date("d.m.Y H:i", strtotime($row['date_taken'])) . "</td>";
        echo "<td class='$score_class'>" . $row['score'] . "%</td>";
        echo "<td><a href='view_result.php?quiz_id=$quiz_id&student_id=" .
$row['student_id'] . "' class='btn-view'>Детальніше</a></td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='4' style='text-align:center; padding:20px; color:#777;'>Ще
ніхто не проходив цей тест.</td></tr>";
}
$stmt->close();
?>
</tbody>
</table>
</main>
</body>
</html>

```

Код файлу «teacher_quizzes.php»:

```

<?php
include 'config.php';
check_auth(1);
$teacher_id = $_SESSION['user_id'];
$subject_id = $_SESSION['teacher_subject_id'];

// Отримка назви предмету вчителя
$subject_name = "Не призначено";
if ($subject_id) {
    $sub_stmt = $db->prepare("SELECT name FROM subjects WHERE id = ?");
    $sub_stmt->bind_param("i", $subject_id);
    $sub_stmt->execute();
    $sub_res = $sub_stmt->get_result();
    if ($row = $sub_res->fetch_assoc()) {
        $subject_name = $row['name'];
    }
    $sub_stmt->close();
}

// Отримка списку класів для фільтру
$classes = [];
$class_stmt = $db->prepare("SELECT DISTINCT class_name FROM quizzes WHERE
teacher_id = ? ORDER BY class_name");
$class_stmt->bind_param("i", $teacher_id);
$class_stmt->execute();
$class_res = $class_stmt->get_result();
while ($row = $class_res->fetch_assoc()) {
    $classes[] = $row['class_name'];
}

```

```

$class_stmt->close();

// Обробка
$filter_class = isset($_GET['filter_class']) ? $_GET['filter_class'] : "";

if (isset($_GET['delete_id'])) {
    $delete_id = (int)$_GET['delete_id'];
    $check = $db->prepare("SELECT id FROM quizzes WHERE id = ? AND teacher_id = ?");
    $check->bind_param("ii", $delete_id, $teacher_id);
    $check->execute();
    if ($check->get_result()->num_rows > 0) {
        $db->query("DELETE FROM quiz_results WHERE quiz_id = $delete_id");
        $db->query("DELETE FROM quizzes WHERE id = $delete_id");
        echo "<script>alert('Тест видалено.');

```

```

padding: 8px 16px;
border-radius: 4px;
font-weight: bold;
font-size: 14px;
transition: background-color 0.2s;
}

nav a:hover { background-color: #5a6268; }

/* АКТИВНА КНОПКА */
nav a.active {
background-color: #495057;
cursor: default;
box-shadow: inset 0 2px 4px rgba(0,0,0,0.2);
}

/* КНОПКА ВИЙТИ */
nav a[href="logout.php"] { background-color: #dc3545; }
nav a[href="logout.php"]:hover { background-color: #c82333; }

/* КОНТРОЛ */
.controls { display: flex; justify-content: space-between; align-items: center; margin: 20px 0; }
.filter-form select { padding: 8px; border-radius: 4px; border: 1px solid #ccc; min-width: 200px; }

/* ТАБЛИЦЯ */
table { width: 100%; border-collapse: collapse; background: #fff; box-shadow: 0 2px 5px rgba(0,0,0,0.1); border-radius: 5px; overflow: hidden; }
th, td { padding: 12px 15px; border-bottom: 1px solid #eee; text-align: left; }
th { background-color: #f8f9fa; color: #333; font-weight: bold; }
tr:last-child td { border-bottom: none; }

/* КНОПКИ ДІЙ В ТАБЛИЦІ */
.btn { padding: 6px 12px; text-decoration: none; border-radius: 4px; color: white; font-size: 13px; display: inline-block; margin-right: 3px; }
.btn-new { background-color: #28a745; padding: 10px 20px; font-size: 15px; font-weight: bold; }
.btn-new:hover { background-color: #218838; }

.btn-edit { background-color: #ffc107; color: #212529; }
.btn-edit:hover { background-color: #e0a800; }

.btn-delete { background-color: #dc3545; }
.btn-delete:hover { background-color: #c82333; }

.btn-results { background-color: #007bff; }
.btn-results:hover { background-color: #0056b3; }
</style>
</head>
<body>
<header>

```

```

<div>
  <h1>Мої Тести</h1>
  <span class="user-info">Вітаємо, <?php echo htmlspecialchars($_SESSION['name']);
?> (Викладач з предмету: <?php echo htmlspecialchars($subject_name); ?>)</span>
</div>
<nav>
  <a href="teacher_quizzes.php" class="active">Мої Тести</a>
  <a href="teacher_grades.php">Оцінки Учнів</a>
  <a href="logout.php">Вийти</a>
</nav>
</header>

<div class="controls">
  <a href="create_quiz.php" class="btn btn-new">+ Створити Новий Тест</a>

  <form class="filter-form" method="GET" style="margin:0;">
    <label>Фільтр по класу: </label>
    <select name="filter_class" onchange="this.form.submit()">
      <option value="">-- Всі класи --</option>
      <?php foreach ($classes as $cls): ?>
        <option value="<?php echo $cls; ?>" <?php echo ($filter_class == $cls) ? 'selected' :
"; ?>>
          <?php echo htmlspecialchars($cls); ?>
        </option>
      <?php endforeach; ?>
    </select>
  </form>
</div>

<main>
  <table>
    <thead>
      <tr>
        <th>Клас</th>
        <th>Тема</th>
        <th>Назва Тесту</th>
        <th>Час</th>
        <th>Питань</th>
        <th>Дії</th>
      </tr>
    </thead>
    <tbody>
      <?php
      $sql = "SELECT
        q.id, q.title, q.time_limit, q.class_name, t.topic_name,
        (SELECT COUNT(*) FROM questions WHERE quiz_id = q.id) as q_count
      FROM quizzes q
      JOIN topics t ON q.topic_id = t.id
      WHERE q.teacher_id = ?";

      if ($filter_class) {
        $sql .= " AND q.class_name = ?";
      }
    </?php
  </tbody>
</table>

```

```

}

$sql .= " ORDER BY q.class_name, t.topic_name";

$stmt = $db->prepare($sql);
if ($filter_class) {
    $stmt->bind_param("is", $teacher_id, $filter_class);
} else {
    $stmt->bind_param("i", $teacher_id);
}

$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td><strong>" . htmlspecialchars($row['class_name']) . "</strong></td>";
        echo "<td>" . htmlspecialchars($row['topic_name']) . "</td>";
        echo "<td>" . htmlspecialchars($row['title']) . "</td>";
        echo "<td>" . htmlspecialchars($row['time_limit']) . " хв.</td>";
        echo "<td>" . $row['q_count'] . "</td>";
        echo "<td>";
        echo "<a href='teacher_quiz_students.php?quiz_id=" . $row['id'] . "' class='btn
btn-results'>Результати</a>";
        echo "<a href='edit_quiz.php?id=" . $row['id'] . "' class='btn btn-edit'>Ред.</a>";
        echo "<a href='teacher_quizzes.php?delete_id=" . $row['id'] . "' class='btn btn-
delete' onclick='return confirm(\"Видалити?\");'>X</a>";
        echo "</td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='6' style='text-align:center; padding:20px;
color:#777;'>Тестів не знайдено.</td></tr>";
}
$stmt->close();
?>
</tbody>
</table>
</main>
</body>
</html>

```

Код файлу «view_result.php»:

```

<?php
include 'config.php';

if (!isset($_SESSION['user_id'])) { header('Location: index.php'); exit; }
$current_user_id = $_SESSION['user_id'];

```

```

$user_role = $_SESSION['role'];

$quiz_id = isset($_GET['quiz_id']) ? (int)$_GET['quiz_id'] : (isset($_GET['id']) ?
(int)$_GET['id'] : 0);
if (!$quiz_id) {
    if ($user_role == 1) header('Location: teacher_quizzes.php');
    else header('Location: student_dashboard.php');
    exit;
}

if ($user_role == 2) {
    $target_student_id = $current_user_id;
    $back_link = "student_dashboard.php";
} elseif ($user_role == 1) {
    $check = $db->prepare("SELECT id FROM quizzes WHERE id = ? AND teacher_id = ?");
    $check->bind_param("ii", $quiz_id, $current_user_id);
    $check->execute();
    if ($check->get_result()->num_rows == 0) die("Доступ заборонено.");

    if (!isset($_GET['student_id'])) die("Помилка: Не обрано студента.");
    $target_student_id = (int)$_GET['student_id'];
    $back_link = "teacher_quiz_students.php?quiz_id=" . $quiz_id;
} else { die("Помилка."); }

$res_query = $db->prepare("SELECT id, score, date_taken, status FROM quiz_results WHERE
student_id = ? AND quiz_id = ?");
$res_query->bind_param("ii", $target_student_id, $quiz_id);
$res_query->execute();
$quiz_result_data = $res_query->get_result()->fetch_assoc();
$res_query->close();

if (!$quiz_result_data) die("Результат не знайдено.");
$result_id = $quiz_result_data['id'];

// Обробка оцінювання вчителем
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['grade_question_id'])) {
    if ($user_role != 1) die("Тільки вчитель може оцінювати.");
    $q_id_to_grade = (int)$_POST['grade_question_id'];
    $points_given = (float)$_POST['points_given'];

    // Оновлення балів за конкретне питання в user_answers
    $upd_ans = $db->prepare("UPDATE user_answers SET points_awarded = ? WHERE
result_id = ? AND question_id = ?");
    $upd_ans->bind_param("dii", $points_given, $result_id, $q_id_to_grade);
    $upd_ans->execute();
    $upd_ans->close();

    // перерахунок балів
    $sum_q = $db->query("SELECT SUM(points_awarded) as total FROM user_answers
WHERE result_id = $result_id");
    $total_earned = $sum_q->fetch_assoc()['total'];
}

```

```

// Максимальний бал
$max_q = $db->query("SELECT SUM(points) as total FROM questions WHERE quiz_id =
$quiz_id");
$max_possible = $max_q->fetch_assoc()['total'];

$new_percent = ($total_earned / $max_possible) * 100;

// Оновлення результату і статусу
$upd_res = $db->prepare("UPDATE quiz_results SET score = ?, status = 'completed' WHERE
id = ?");
$upd_res->bind_param("di", $new_percent, $result_id);
$upd_res->execute();

header("Location: view_result.php?quiz_id=$quiz_id&student_id=$target_student_id");
exit;
}

$quiz_info = $db->query("SELECT title FROM quizzes WHERE id = $quiz_id")-
>fetch_assoc();
$student_info = $db->query("SELECT name FROM users WHERE id = $target_student_id")-
>fetch_assoc();

// Завантаження відповіді студента
$user_answers = [];
$ua_query = $db->prepare("SELECT question_id, chosen_answer_id FROM user_answers
WHERE result_id = ?");
$ua_query->bind_param("i", $result_id);
$ua_query->execute();
$ua_res = $ua_query->get_result();
while ($row = $ua_res->fetch_assoc()) {
    if($row['chosen_answer_id']) $user_answers[$row['question_id']][] =
$row['chosen_answer_id'];
}
$ua_query->close();
?>
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Результат тесту</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4; color: #333;
    }
        header { background-color: #fff; padding: 15px 20px; border-radius: 5px; border-bottom:
2px solid #ccc; margin-bottom: 20px; display: flex; justify-content: space-between; align-items:
center; box-shadow: 0 2px 5px rgba(0,0,0,0.05); }
        header h1 { margin: 0; font-size: 24px; }
        .back-btn { text-decoration: none; color: #007bff; font-weight: bold; padding: 8px 15px;
border: 1px solid #007bff; border-radius: 4px; }
        .student-name-label { display: block; font-size: 0.9em; color: #555; margin-top: 5px; }
        .container { max-width: 800px; margin: 0 auto; }

```

```

        .summary-box { background: #fff; padding: 25px; border-radius: 8px; border: 1px solid
#ccc; text-align: center; margin-bottom: 30px; box-shadow: 0 4px 10px rgba(0,0,0,0.05); }
        .summary-score { font-size: 3em; color: #28a745; font-weight: bold; margin: 10px 0; }
        .status-pending { color: orange; font-weight: bold; }
        .question-card { background: #fff; padding: 20px; border-radius: 8px; border: 1px solid
#ddd; margin-bottom: 20px; box-shadow: 0 2px 5px rgba(0,0,0,0.05); }
        .question-text { font-size: 1.1em; font-weight: bold; margin-bottom: 15px; border-bottom:
1px solid #eee; padding-bottom: 10px; }
        .points { font-size: 0.8em; color: #777; font-weight: normal; float: right; background:
#f8f9fa; padding: 2px 8px; border-radius: 10px; border: 1px solid #eee; }
        .answer-item { padding: 12px; margin-bottom: 8px; border-radius: 4px; border: 1px solid
#eee; display: flex; justify-content: space-between; align-items: center; background-color: #fff; }
        .correct-answer { background-color: #d4edda; border-color: #c3e6cb; color: #155724; }
        .wrong-choice { background-color: #f8d7da; border-color: #f5c6cb; color: #721c24; }
        .user-choice-marker { font-weight: bold; font-size: 0.85em; text-transform: uppercase;
margin-left: 10px; }
        .text-answer-box { background: #f9f9f9; padding: 15px; border: 1px solid #ccc; border-
radius: 4px; font-style: italic; }
    </style>
</head>
<body>

<header>
    <div>
        <h1>Результат: <?php echo htmlspecialchars($quiz_info['title']); ?></h1>
        <?php if($user_role == 1): ?>
            <span class="student-name-label">Учень: <strong><?php echo
htmlspecialchars($student_info['name']); ?></strong></span>
            <?php endif; ?>
        </div>
        <a href="<?php echo $back_link; ?>" class="back-btn">← Назад</a>
    </header>

<div class="container">
    <div class="summary-box">
        <div>Загальний результат</div>
        <div class="summary-score"><?php echo $quiz_result_data['score']; ?>%</div>
        <?php if($quiz_result_data['status'] == 'pending'): ?>
            <div class="status-pending">Очікує перевірки вчителем</div>
        <?php endif; ?>
        <div style="color: #777; margin-top: 5px;">Дата: <?php echo date("d.m.Y H:i",
strtotime($quiz_result_data['date_taken'])); ?></div>
    </div>

    <?php
    $q_stmt = $db->prepare("SELECT id, question_text, points, type FROM questions WHERE
quiz_id = ? ORDER BY id ASC");
    $q_stmt->bind_param("i", $quiz_id);
    $q_stmt->execute();
    $questions = $q_stmt->get_result();

    $count = 1;

```

```

while ($q = $questions->fetch_assoc()) {
    echo "<div class='question-card'>";
    echo "<div class='question-text'>{$count} . " . htmlspecialchars($q['question_text']) .
        " <span class='points'>" . $q['points'] . " б.</span></div>";

    if ($q['type'] == 'text') {
        // Текстова відповідь
        $txt_q = $db->query("SELECT text_answer, points_awarded FROM user_answers
WHERE result_id = $result_id AND question_id = " . $q['id']);
        $txt_row = $txt_q->fetch_assoc();
        $student_text = $txt_row ? $txt_row['text_answer'] : "";
        $awarded = $txt_row ? $txt_row['points_awarded'] : 0;

        echo "<div class='text-answer-box'>" . nl2br(htmlspecialchars($student_text)) . "</div>";

        if ($user_role == 1) {
            // Форма оцінювання
            echo "<form method='POST' style='margin-top:15px; display:flex; gap:10px; align-
items:center; background:#e9ecef; padding:10px; border-radius:4px;'>";
            echo "<input type='hidden' name='grade_question_id' value='" . $q['id'] . "'>";
            echo "<label><strong>Оцінка:</strong></label>";
            echo "<input type='number' name='points_given' value='$awarded' min='0' max='" .
            $q['points'] . "' step='0.1' style='width:60px; padding:5px;'>";
            echo " / " . $q['points'];
            echo "<button type='submit' style='margin-left:auto; background:#28a745;
color:white; border:none; padding:6px 12px; cursor:pointer; border-
radius:4px;'>Зберегти</button>";
            echo "</form>";
        } else {
            // Вивід оцінки
            echo "<div style='margin-top:10px; color:#28a745; font-weight:bold;'>Оцінка
вчителя: $awarded / " . $q['points'] . "</div>";
        }
    } else {
        // Тестова відповідь
        $a_stmt = $db->prepare("SELECT id, answer_text, is_correct FROM answers WHERE
question_id = ? ORDER BY id ASC");
        $a_stmt->bind_param("i", $q['id']);
        $a_stmt->execute();
        $answers = $a_stmt->get_result();
        $user_choices = isset($user_answers[$q['id']]) ? $user_answers[$q['id']] : [];

        while ($a = $answers->fetch_assoc()) {
            $class = ""; $marker = "";
            $isSelected = in_array($a['id'], $user_choices);

            if ($a['is_correct']) {
                $class = "correct-answer";
                $marker = $isSelected ? "✓ Ви обрали (Вірно)" : "✓ Правильна відповідь";
            } elseif ($isSelected) {
                $class = "wrong-choice";
            }
        }
    }
}

```

```

        $marker = "X Ви обрали (Невірно)";
    }

    echo "<div class='answer-item $class'>";
    echo "<span>" . htmlspecialchars($a['answer_text']) . "</span>";
    if ($marker) echo "<span class='user-choice-marker'>$marker</span>";
    echo "</div>";
}
}
echo "</div>";
$count++;
}
$q_stmt->close();
?>
</div>
</body>
</html>

```

Код файлу «admin_dashboard.php»:

```

<?php
include 'config.php';
check_auth(3); // Доступ тільки для Адміністратора

// Логіка видалення
if (isset($_GET['delete_id'])) {
    $del_id = (int)$_GET['delete_id'];

    // не можна видалити самого себе
    if ($del_id == $_SESSION['user_id']) {
        echo "<script>alert('Не можна видалити власний акаунт!');
location.href='admin_dashboard.php';</script>";
        exit;
    }

    // Дізнаємось роль користувача перед видаленням
    $role_q = $db->query("SELECT role FROM users WHERE id = $del_id");

    if ($role_q->num_rows > 0) {
        $target_role = $role_q->fetch_assoc()['role'];

        // Очищення даних
        if ($target_role == 1) { // Якщо це Вчитель
            // Знаходимо всі його тести
            $quizzes = $db->query("SELECT id FROM quizzes WHERE teacher_id = $del_id");
            while ($q = $quizzes->fetch_assoc()) {
                $qid = $q['id'];

                // Видаляємо результати учнів за ці тести
                $db->query("DELETE FROM quiz_results WHERE quiz_id = $qid");
            }
        }
    }
}

```

```

$db->query("DELETE FROM questions WHERE quiz_id = $qid");

}
// Видалення тестів
$db->query("DELETE FROM quizzes WHERE teacher_id = $del_id");

} elseif ($target_role == 2) {
    // Для учня
    // Видаляємо загальні результати
    $db->query("DELETE FROM quiz_results WHERE student_id = $del_id");
    // Видаляємо детальні відповіді на питання
}

// Фінальне видалення користувача з системи
if ($db->query("DELETE FROM users WHERE id = $del_id")) {
    header("Location: admin_dashboard.php");
    exit;
} else {
    echo "Помилка видалення: " . $db->error;
}
}
}

// Отримання списку користувачів
$sql = "SELECT users.*, subjects.name as subject_name
FROM users
LEFT JOIN subjects ON users.subject_id = subjects.id
ORDER BY role DESC, name ASC";
$users = $db->query($sql);
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Панель Адміністратора</title>
    <style>
        body { font-family: Arial, sans-serif; padding: 20px; background: #f4f4f4; color: #333; }

        /* Шапка */
        .header {
            display: flex; justify-content: space-between; align-items: center;
            background: #fff; padding: 15px 20px; border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
        }
        h1 { margin: 0; font-size: 24px; }
        .logout-btn { color: #dc3545; text-decoration: none; font-weight: bold; }
        .logout-btn:hover { text-decoration: underline; }

        /* Кнопки керування */
        .controls { margin: 20px 0; display: flex; gap: 10px; }

```

```

.btn { padding: 10px 15px; text-decoration: none; color: white; border-radius: 4px; font-size: 14px; display: inline-block; font-weight: bold; transition: background 0.2s; }

.btn-add { background: #28a745; } /* Зелена */
.btn-add:hover { background: #218838; }

.btn-subjects { background: #17a2b8; } /* Блакитна */
.btn-subjects:hover { background: #138496; }

/* Таблиця */
table { width: 100%; border-collapse: collapse; background: #fff; border-radius: 5px; overflow: hidden; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
th, td { padding: 12px 15px; text-align: left; border-bottom: 1px solid #eee; }
th { background-color: #343a40; color: white; }
tr:hover { background-color: #f9f9f9; }

// Бейджі ролей
.badge { padding: 4px 8px; border-radius: 4px; color: white; font-size: 0.8em; font-weight: normal; }
.badge-admin { background: #dc3545; } /* Червоний */
.badge-teacher { background: #007bff; } /* Синій */
.badge-student { background: #28a745; } /* Зелений */

/* Кнопки в таблиці */
.btn-sm { padding: 5px 10px; font-size: 12px; font-weight: normal; }
.btn-edit { background: #ffc107; color: #333; }
.btn-edit:hover { background: #e0a800; }
.btn-del { background: #dc3545; }
.btn-del:hover { background: #c82333; }
</style>
</head>
<body>

<div class="header">
  <h1>Керування системою</h1>
  <div>
    Ви увійшли як: <b><?php echo htmlspecialchars($_SESSION['name']); ?></b> |
    <a href="logout.php" class="logout-btn">Вийти</a>
  </div>
</div>

<div class="controls">
  <a href="admin_edit_user.php" class="btn btn-add">+ Додати нового користувача</a>
  <a href="admin_subjects.php" class="btn btn-subjects">Керування предметами</a>
</div>

<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>ПІБ</th>
      <th>Логін</th>
    </tr>
  </thead>
</table>

```

```

        <th>Роль</th>
        <th>Деталі (Клас / Предмет)</th>
        <th>Дії</th>
    </tr>
</thead>
<tbody>
    <?php while($u = $users->fetch_assoc()): ?>
    <tr>
        <td><?php echo $u['id']; ?></td>
        <td><?php echo htmlspecialchars($u['name']); ?></td>
        <td><?php echo htmlspecialchars($u['login']); ?></td>
        <td>
            <?php
            if ($u['role'] == 3) echo "<span class='badge badge-admin'>Адмін</span>";
            elseif ($u['role'] == 1) echo "<span class='badge badge-teacher'>Вчитель</span>";
            elseif ($u['role'] == 2) echo "<span class='badge badge-student'>Учень</span>";
            ?>
        </td>
        <td>
            <?php
            if ($u['role'] == 2) {
                echo "Клас: <strong>" . htmlspecialchars($u['class_name']) . "</strong>";
            } elseif ($u['role'] == 1) {
                echo "Предмет: <strong>" . htmlspecialchars($u['subject_name'] ?? 'Не вказано')
                . "</strong>";
            } else {
                echo "-";
            }
            ?>
        </td>
        <td>
            <a href="admin_edit_user.php?id=<?php echo $u['id']; ?>" class="btn btn-sm btn-
            edit">Ред.</a>

            <?php if ($u['id'] != $_SESSION['user_id']): ?>
                <a href="?delete_id=<?php echo $u['id']; ?>" class="btn btn-sm btn-del"
                onclick="return confirm('Ви впевнені? \n\nЯкщо це ВЧИТЕЛЬ: будуть видалені всі його
                тести та результати учнів.\nЯкщо це УЧЕНЬ: будуть видалені всі його
                оцінки.');">Вид.</a>
            <?php endif; ?>
        </td>
    </tr>
    <?php endwhile; ?>
</tbody>
</table>

</body>
</html>

```

Код файлу «admin_edit_user.php»:

```

<?php
include 'config.php';
check_auth(3); // Доступ тільки для адміна

$хid = isset($_GET['id']) ? (int)$_GET['id'] : 0;
$хuser = ['name'=>'', 'login'=>'', 'role'=>2, 'class_name'=>'', 'subject_id'=>0];
$хis_edit = false;

// дістаємо дані для редагування
if ($хid) {
    $хis_edit = true;
    $хstmt = $хdb->prepare("SELECT * FROM users WHERE id = ?");
    $хstmt->bind_param("i", $хid);
    $хstmt->execute();
    $хuser = $хstmt->get_result()->fetch_assoc();
    if (!$хuser) die("Користувача не знайдено");
}

// Отримуємо список предметів для форми
$хsubjects = $хdb->query("SELECT * FROM subjects ORDER BY name");

// ОБРОБКА ФОРМИ
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $хname = $_POST['name'];
    $хlogin = $_POST['login'];
    $хrole = (int)$_POST['role'];
    // Якщо роль не учень, клас очищаємо. Якщо не вчитель, предмет очищаємо.
    $хclass_name = ($хrole === 2) ? $_POST['class_name'] : NULL;
    $хsubject_id = ($хrole === 1) ? (int)$_POST['subject_id'] : NULL;
    $хpassword = $_POST['password'];

    if ($хis_edit) {
        // Оновлення
        $хsql = "UPDATE users SET name=?, login=?, role=?, class_name=?, subject_id=?
WHERE id=?";
        $хstmt = $хdb->prepare($хsql);
        $хstmt->bind_param("ssisi", $хname, $хlogin, $хrole, $хclass_name, $хsubject_id, $хid);
        $хstmt->execute();

        // Оновлення пароля тільки якщо введено новий
        if (!empty($хpassword)) {
            $хhash = password_hash($хpassword, PASSWORD_DEFAULT);
            $хdb->query("UPDATE users SET password='$хhash' WHERE id=$хid");
        }
    } else {
        // Створення
        if (empty($хpassword)) die("Помилка: Пароль обов'язковий для нового користувача.");
        $хhash = password_hash($хpassword, PASSWORD_DEFAULT);
    }
}

```

```

    $sql = "INSERT INTO users (name, login, password, role, class_name, subject_id)
VALUES (?, ?, ?, ?, ?, ?)";
    $stmt = $db->prepare($sql);
    $stmt->bind_param("ssssi", $name, $login, $hash, $role, $class_name, $subject_id);
    $stmt->execute();
}

// Повернення до списку
header("Location: admin_dashboard.php");
exit;
}
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title><?php echo $is_edit ? 'Редагування' : 'Новий'; ?> користувач</title>
    <style>
        body { font-family: Arial, sans-serif; background: #f4f4f4; padding: 20px; display: flex;
justify-content: center; }
        .form-container { background: #fff; padding: 30px; border-radius: 5px; width: 400px; box-
shadow: 0 4px 10px rgba(0,0,0,0.1); }
        h2 { margin-top: 0; text-align: center; color: #333; }

        .form-group { margin-bottom: 15px; }
        label { display: block; margin-bottom: 5px; font-weight: bold; color: #555; }
        input[type="text"], select { width: 100%; padding: 10px; border: 1px solid #ccc; border-
radius: 4px; box-sizing: border-box; }

        button { width: 100%; padding: 12px; background: #28a745; color: white; border: none;
border-radius: 4px; font-size: 16px; cursor: pointer; margin-top: 10px; }
        button:hover { background: #218838; }

        .cancel-btn { display: block; text-align: center; margin-top: 15px; color: #007bff; text-
decoration: none; }
    </style>
</head>
<body>

    <div class="form-container">
        <h2><?php echo $is_edit ? 'Редагування' : 'Створення'; ?></h2>

        <form method="POST">
            <div class="form-group">
                <label>ПІБ:</label>
                <input type="text" name="name" value="<?php echo htmlspecialchars($user['name']);
?>" required placeholder="Іванов Іван">
            </div>

            <div class="form-group">
                <label>Логін:</label>

```

```

        <input type="text" name="login" value="<?php echo htmlspecialchars($user['login']);
?>" required>
    </div>

    <div class="form-group">
        <label>Пароль <?php if($is_edit) echo '<small style="font-weight:normal">(залиште
пустим, щоб не змінювати)</small>'; ?>:</label>
        <input type="text" name="password" placeholder="Введіть пароль">
    </div>

    <div class="form-group">
        <label>Роль:</label>
        <select name="role" id="roleSelect" onchange="toggleFields()">
            <option value="2" <?php if($user['role']==2) echo 'selected'; ?>>Учень</option>
            <option value="1" <?php if($user['role']==1) echo 'selected'; ?>>Вчитель</option>
            <option value="3" <?php if($user['role']==3) echo 'selected';
?>>Адміністратор</option>
        </select>
    </div>

    <div class="form-group" id="studentFields">
        <label>Клас (для учня):</label>
        <input type="text" name="class_name" value="<?php echo
htmlspecialchars($user['class_name'] ?? ''); ?>" placeholder="Наприклад: 10-А">
    </div>

    <div class="form-group" id="teacherFields" style="display:none;">
        <label>Предмет (для вчителя):</label>
        <select name="subject_id">
            <option value="0">-- Не обрано --</option>
            <?php while($s = $subjects->fetch_assoc()): ?>
                <option value="<?php echo $s['id']; ?>" <?php if(($user['subject_id']??0) ==
$s['id']) echo 'selected'; ?>>
                <?php echo htmlspecialchars($s['name']); ?>
            </option>
            <?php endwhile; ?>
        </select>
    </div>

    <button type="submit">Зберегти</button>
    <a href="admin_dashboard.php" class="cancel-btn">Скасувати</a>
</form>
</div>

<script>
// Скрипт для показу/приховування полів залежно від ролі
function toggleFields() {
    let role = document.getElementById('roleSelect').value;
    let studentDiv = document.getElementById('studentFields');
    let teacherDiv = document.getElementById('teacherFields');

    if (role == '2') { // Учень

```

```

        studentDiv.style.display = 'block';
        teacherDiv.style.display = 'none';
    } else if (role == '1') { // Вчитель
        studentDiv.style.display = 'none';
        teacherDiv.style.display = 'block';
    } else { // Адмін
        studentDiv.style.display = 'none';
        teacherDiv.style.display = 'none';
    }
}
// Запуск при завантаженні сторінки, щоб встановити правильний стан при
редагуванні
toggleFields();
</script>

</body>
</html>

```

Код файлу «admin_subjects.php»:

```

<?php
include 'config.php';
check_auth(3); // Тільки для адміну

// Додавання предмету
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['new_subject'])) {
    $name = trim($_POST['subject_name']);
    if (!empty($name)) {
        // Перевірка чи такий вже є
        $check = $db->query("SELECT id FROM subjects WHERE name = '$name'");
        if ($check->num_rows == 0) {
            $stmt = $db->prepare("INSERT INTO subjects (name) VALUES (?)");
            $stmt->bind_param("s", $name);
            $stmt->execute();
            $stmt->close();
        } else {
            $error = "Такий предмет вже існує!";
        }
    }
}

// Видалення предмету
if (isset($_GET['delete_id'])) {
    $del_id = (int)$_GET['delete_id'];
    // Спочатку перевіряємо, чи не прив'язаний предмет до вчителів або тестів (це важливо
для цілісності)
    $db->query("DELETE FROM subjects WHERE id = $del_id");
    header("Location: admin_subjects.php");
    exit;
}

```

```

}

$subjects = $db->query("SELECT * FROM subjects ORDER BY name ASC");
?>

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Керування предметами</title>
  <style>
    body { font-family: Arial, sans-serif; padding: 20px; background: #f4f4f4; display: flex;
flex-direction: column; align-items: center; }
    .container { background: #fff; padding: 30px; border-radius: 5px; width: 500px; box-
shadow: 0 4px 10px rgba(0,0,0,0.1); }
    h2 { margin-top: 0; text-align: center; }

    form { display: flex; gap: 10px; margin-bottom: 20px; border-bottom: 1px solid #eee;
padding-bottom: 20px; }
    input[type="text"] { flex-grow: 1; padding: 10px; border: 1px solid #ccc; border-radius:
4px; }
    button { padding: 10px 20px; background: #28a745; color: white; border: none; border-
radius: 4px; cursor: pointer; }
    button:hover { background: #218838; }

    table { width: 100%; border-collapse: collapse; }
    td { padding: 10px; border-bottom: 1px solid #eee; }
    .del-btn { color: #dc3545; text-decoration: none; font-weight: bold; float: right; }
    .back-link { display: block; margin-bottom: 15px; text-decoration: none; color: #007bff; }
  </style>
</head>
<body>

  <div class="container">
    <a href="admin_dashboard.php" class="back-link">← Назад в адмін-панель</a>
    <h2>Список предметів</h2>

    <?php if(isset($error)) echo "<p style='color:red'>$error</p>"; ?>

    <form method="POST">
      <input type="hidden" name="new_subject" value="1">
      <input type="text" name="subject_name" placeholder="Назва нового предмету (напр.
Географія)" required>
      <button type="submit">Додати</button>
    </form>

    <table>
      <?php while($s = $subjects->fetch_assoc()): ?>
        <tr>
          <td><?php echo htmlspecialchars($s['name']); ?></td>
          <td>

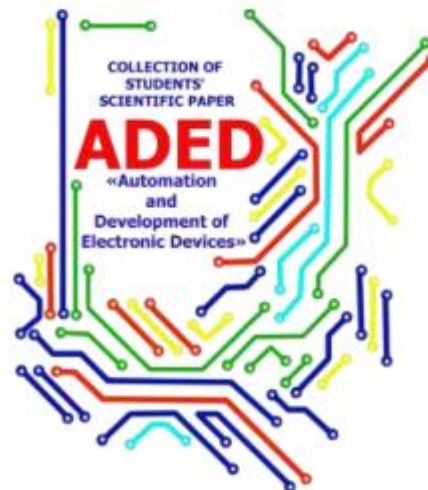
```

```
        <a href="?delete_id=<?php echo $$['id']; ?>" class="del-btn" onclick="return
confirm('Видалити цей предмет?');">x</a>
    </td>
</tr>
<?php endwhile; ?>
</table>
</div>

</body>
</html>
```

ДОДАТОК Б
Апробація наукових результатів досліджень

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК
студентських наукових статей
«Автоматизація та приладобудування»
«Automation and Development of Electronic Devices»
ADED-2024
(Випуск 2)
[електронне видання]

Харків 2024

АНАЛІЗ МЕТОДІВ І ФУНКЦІЙ ЗАХИСТУ ДАНИХ ДЛЯ РЕСУРСІВ ДИСТАНЦІЙНОГО НАВЧАННЯ

Обривко Є.В.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: yevhen.obryvko@nure.ua

Анотація: В роботі розглянуто актуальне питання підвищення надійності та захисту персональних даних користувачів та закладу освіти для ресурсу дистанційного навчання. Проведено аналіз існуючих методів захисту даних на веб-сторінках, їх ефективності, актуальності та особливостей застосування, також проаналізовано методи захист паролів та їх надійність.

Ключові слова: захист даних, шифрування, ресурс, дистанційне навчання, навантаження, сервер

ANALYSIS OF DATA PROTECTION METHODS AND FUNCTIONS FOR DISTANCE LEARNING RESOURCES

Obryvko E.V.

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky ave. 14

E-mail: yevhen.obryvko@nure.ua

Abstract. The paper considers the current issue of increasing the reliability and protection of personal data of users and educational institutions for a distance learning resource. An analysis of existing methods of data protection on web pages, their effectiveness, relevance and application features, and methods of password protection and their reliability are also analyzed.

Keywords: data protection, encryption, resource, distance learning, load, server

АКТУАЛЬНІСТЬ РОБОТИ. З переходом до дистанційних форм навчання питання безпеки даних користувачів набуло ще більшої важливості. Освітні платформи сьогодні об'єднують тисячі користувачів, від студентів до викладачів, що працюють із персональними акаунтами та конфіденційною інформацією [1].

Системи дистанційного навчання зберігають значні обсяги особистої та академічної інформації, яка піддається ризикам зловживань і витоків. Уразливості в захисті даних можуть призвести до серйозних наслідків, як-от витік даних, фінансові втрати або навіть порушення репутації освітнього закладу [2].

Також важливим фактором є відповідність вимогам законодавства про захист даних. Урахування вимог цих законодавств робить впровадження надійних систем захисту [3] ще більш актуальним.

Внаслідок швидкого розвитку технологій та методів атак зловмисники постійно вдосконалюють свої методи для отримання доступу до конфіденційних даних. На додачу, розвиток технологій штучного інтелекту та машинного навчання дозволяє створювати складні атаки, які важко розпізнати без відповідного захисту. У зв'язку з цим аналіз та оновлення методів захисту даних є необхідними для підтримання безпеки освітніх платформ [1].

Отже при розробці подібних системи безпеки даних та захисту персональної інформації має приділятися велика увага з урахування усіх вище перелічених факторів [4].

ВСТУП. Для ефективного забезпечення безпеки важливо розуміти, з якими загрозами стикаються освітні платформи. Основні загрози включають [5-7]:

Витік даних – уразливості в системах безпеки, які можуть призвести до несанкціонованого доступу до персональної інформації, наприклад, облікових записів, результатів навчання або особистих даних студентів.

Фішинг – метод отримання доступу до конфіденційної інформації шляхом шахрайства. Фішингові атаки можуть приймати вигляд електронних листів, що імітують офіційні запити від навчальних платформ, які просять надати логін або пароль.

Атаки «людина посередині» (MITM) – перехоплення даних між сервером та клієнтом, що дозволяє зловмисникам отримати доступ до інформації, яка передається під час роботи з платформою.

DdoS-атаки – масове навантаження на сервер, що може призвести до повної або часткової втрати доступу до освітньої платформи.

МАТЕРІАЛИ ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ.

Шифрування даних.

Шифрування даних є базовим інструментом для захисту конфіденційної інформації. Розглянемо основні методи шифрування:

1) симетричне шифрування (AES) – один ключ використовується для шифрування та розшифрування інформації. AES забезпечує високу швидкість обробки даних і є ефективним для великих обсягів;

2) асиметричне шифрування (RSA) – використовує пару ключів: відкритий для шифрування та закритий для розшифрування. RSA забезпечує високий рівень захисту, однак потребує значних обчислювальних ресурсів, тому його ефективно використовувати лише для захисту невеликих обсягів критичної інформації.

Формули шифрування на основі відкритого ключа:

$$C = E(K_{pub}, P)$$

де C – зашифрований текст, K_{pub} – відкритий ключ, P – відкритий текст.

Багатофакторна автентифікація.

Одним із найбільш дієвих методів захисту доступу до акаунтів є багатофакторна автентифікація (2FA). Ця система включає використання додаткового коду, який генерується або відправляється через інший канал (наприклад, телефон або електронну пошту), що знижує ризик несанкціонованого доступу. До методів автентифікації належать:

1) TOTP (Time-based One-Time Password) – одноразові паролі з обмеженим часом дії;

2) біометричні дані – відбитки пальців, розпізнавання обличчя, сканування сітківки тощо, що надають найвищий рівень захисту.

Захист від DdoS-атак.

DdoS-атака – напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена. Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає настільки повільно, що стає фактично недоступним.

Для запобігання відмові в обслуговуванні через масові DdoS-атаки ефективними є:

1) балансування навантаження: розподіл запитів на декілька серверів для забезпечення стабільної роботи платформи;

2) фільтрування трафіку на рівні мережі для зменшення впливу небажаних запитів.

Хешування паролів.

Хешування – це процес перетворення пароля на унікальний хеш-код, який зберігається в базі даних замість самого пароля. Коли користувач вводить свій пароль, система хешує його і порівнює з наявним хешем. Якщо хеші збігаються, доступ дозволяється. Найпоширенішими алгоритмами для хешування є SHA-256, bcrypt, Argon2, які забезпечують різні рівні надійності.

1) bcrypt – алгоритм хешування, стійкий до атак підбору, оскільки включає «сіль» (salt) і адаптивний параметр, що уповільнює обчислення;

2) Argon2 – більш сучасний алгоритм хешування, оптимізований для захисту від атак на графічних процесорах, з додатковими параметрами стійкості.

Формула для хешування пароля з використанням «солі»:

$$H = \text{hash}(\text{password} + \text{salt})$$

де H – хеш, password – пароль користувача, salt – випадкове значення, що додається до кожного паролю користувача.

У таблиці 1 нижче наведено порівняння методів шифрування та захисту даних і підвищення безпеки системи з точки зору ефективності використання, ефективності та особливостей.

Таблиця 1 – Порівняння основних методів шифрування, захисту та підвищення безпеки системи.

Метод	Ефективність	Вартість впровадження	Особливості
AES шифрування	Висока	Помірна	Підходить для великих обсягів даних
RSA шифрування	Висока	Висока	Ефективний для критичних даних
Багатофакторна автентифікація (2FA)	Висока	Висока	Захищає від крадіжки облікових записів
SSL/TLS сертифікати	Висока	Помірна	Забезпечує захист від MITM-атак
Хешування паролів (bcrypt, Argon2)	Висока	Низька	Забезпечує надійність паролів
Контроль доступу	Висока	Низька	Дозволяє контролювати права доступу користувачів
Балансування навантаження	Середня	Висока	Зменшує швидкість обробки запитів
Фільтрування трафіку	Середня	Висока	Знижує ризик небажаних запитів

Порівняння основних методи шифрування AES та RSA. Оскільки у системі дистанційної освіти передбачено використання великих обсягів даних, таких як навчальні матеріали, оцінки, персональні дані та паролі, то основними критеріями при виборі методу шифрування будуть захист критичних даних та стабільність і швидкість роботи при великих об'ємах інформації. Тому порівняємо два найбільш доречних у цих умовах методи AES шифрування та RSA шифрування при різних обсягах інформації (рис. 1).

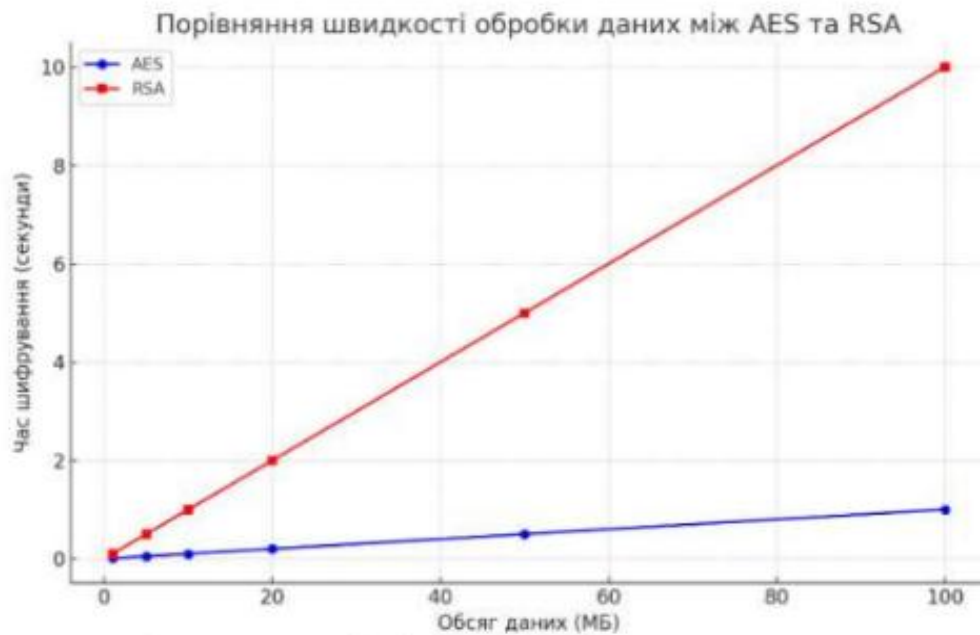


Рисунок 1 – Порівняння швидкості обробки даних між алгоритмами AES та RSA при різних обсягах інформації

На графіку видно, що AES має значно вищу швидкість обробки даних порівняно з RSA, особливо при більших обсягах даних, що робить його більш ефективним для великих інформаційних масивів.

Оскільки у системі дистанційної освіти передбачено використання великих обсягів даних, таких як навчальні матеріали, оцінки, персональні дані та паролі то доцільніше з точки зору вартості впровадження та ефективності буде використовувати метод AES шифрування.

Для ілюстрації основних етапів обробки інформації в системі дистанційного навчання та подальшого визначення для них відповідних методів захисту, створимо схему потоку даних у системі (рис. 2).



Рисунок 2 – Схема потоку даних у системі дистанційного навчання

Як бачимо на рис. 2, у системі є 5 основних етапів обробки інформації:

- Користувачі (студенти, викладачі);
- Передача даних (мережа);
- Сервер додатку;
- База даних;
- Адміністрація.

Для забезпечення найвищого рівня безпеки даних, кожному з етапів обробки інформації бажано мати свій метод захисту для системи в нашому випадку, базуючись на попередньому аналізі методів та засобів, можна зробити наступний розподіл етапів потоку даних та засобів захисту для них (рис. 3).

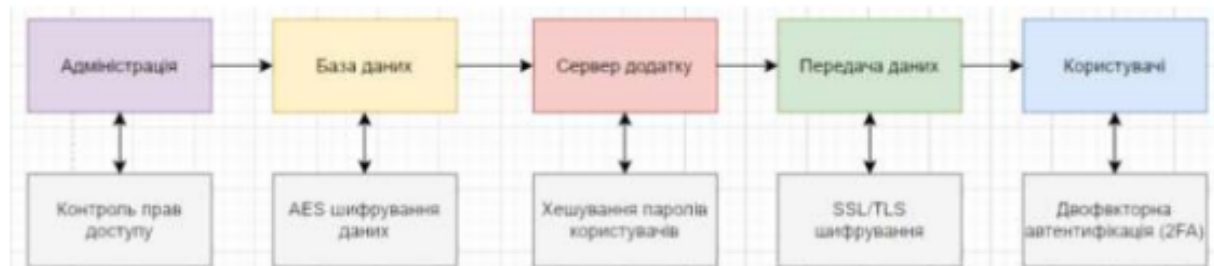


Рисунок 3 – Розподіл етапів потоку даних та засобів захисту

ВИСНОВКИ. Для поставлених задач системи більш за все підходить метод AES шифрування, проте поєднання методів шифрування, багатофакторної автентифікації та захисту від DDoS-атак дозволяє досягти високого рівня безпеки для ресурсів дистанційного навчання. Доцільно впроваджувати багатоварову систему захисту, яка охоплює шифрування даних, автентифікацію користувачів і систему фільтрації трафіку. Використання штучного інтелекту для аналізу загроз у реальному часі може стати наступним кроком у забезпеченні безпеки освітніх платформ. Інтеграція інтелектуальних систем моніторингу дозволить не тільки реагувати на кіберзагрози, а й прогнозувати їх.

ВИКОРИСТАНА ЛІТЕРАТУРА.

1. Anees, A., Hussain, I., Khokhar, U. M., Ahmed, F., & Shaukat, S. (2022). Machine learning and applied cryptography. *Security & Communication Networks*.
2. Віртуальне навчальне середовище: Вікіпедія. URL: <https://uk.wikipedia.org/wiki/DoS-атака> (дата звернення: 27.10.2024).
3. Tibouchi, M., & Wang, X. (2023). Applied cryptography and network security. In *Proceedings of the 21st International Conference on ACNS* (Vol. 13906).
4. William Stallings. *Cryptography and Network Security: Principles and Practice* // Prentice Hall. 1999. – С. 118-154.
5. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2021). Handbook of applied cryptography. *Instructor, 202101*.
6. Шифрування. Типи і алгоритми: hostpro.ua. URL: <https://hostpro.ua/wiki/ua/security/encryption-types-algorithms/> (дата звернення: 29.10.2024).
7. Хешування паролів: використання солі та bcrypt: drukarnia.com.ua. URL: <https://drukarnia.com.ua/articles/kheshuvannya-paroliv-vikoristannya-soli-ta-bcrypt> fsme-#heading-3-350 (дата звернення: 29.10.2024).

8. Yevsieiev V. Ecosystem Model of the Concept of Industry 5.0 / V. Yevsieiev // Digital innovation & sustainable development 2024 : Proceedings of I-st International Conference, November 15, 2024. – Kharkiv, 2024. – P. 12-13.
9. Moiseev, M., Maksymova, S., Yevsieiev, V., & Alkhalailah, A. (2024). Program Algorithm for Monitoring System Development. *Journal of universal science research*, 2(7), 33-43.
10. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks. *Computational Intelligence and Neuroscience*, 2022, Article ID 5866922. <https://doi.org/10.1155/2022/5866922>.
11. Lyashenko, V., Abu-Jassar, A.T., Yevsieiev, V., Maksymova, S. Automated Monitoring and Visualization System in Production, *Int. Res. J. Multidiscip. Technovation*, 5(6) 2023 09-18. <https://doi.org/10.54392/irjmt2362>
12. Amer Abu-Jassar, Vladyslav Yevsieiev, & Svitlana Maksymova. (2024). The Optical Flow Method and Graham's Algorithm Implementation Features for Searching for the Object Contour in the Mobile Robot's Workspace. *Journal of Universal Science Research*, 2(3), 64–75.
13. Vladyslav Yevsieiev, Samariddin, S. M., Nikolay Starodubtsev, & Amer Abu-Jassar. (2024). ACTIVE CONTOURS METHOD IMPLEMENTATION FOR OBJECTS SELECTION IN THE MOBILE ROBOT'S WORKSPACE. *Journal of Universal Science Research*, 2(2), 135–145.
14. Svitlana Maksymova, Vladyslav Yevsieiev, & Ahmad Alkhalailah. (2024). The Monitoring System Architecture Development. *Journal of Universal Science Research*, 2(1), 69–79.
15. Andrii Bondariev, Svitlana Maksymova, & Vladyslav Yevsieiev. (2023). Automated Monitoring System Development for Equipment Modernization. *Journal of Universal Science Research*, 1(11), 6–16. Retrieved from <https://universalpublishings.com/index.php/jusr/article/view/2484>
16. Yevsieiev, V. Comparative Analysis of the Characteristics of Mobile Robots and Collaboration Robots Within INDUSTRY 5.0. / V. Yevsieiev, D. Gurin // In the VI International Scientific and Theoretical Conference, September 8, 2023. Chicago, USA. P.92-94
17. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. *Technical science research in Uzbekistan*, 2(2), 115- 124.
18. Gurin, D., & et al. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. *Multidisciplinary Journal of Science and Technology*, 4(8), 36-44.
19. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. *Technical Science Research In Uzbekistan*, 2(3), 103-112
20. Gurin, D., & et al. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. *Multidisciplinary Journal of Science and Technology*, 4(8), 5-12.
21. Yevsieiev, V., & Uskov, S. (2024). *Development of the Layout Concept of a Small-Dimensioned Mobile Robot With Increased Accessibility* (Doctoral dissertation, International Scientific Unity).
22. Yevsieiev, V., & Gurin, D. (2024). *New Concepts of Human Interactions and Collaborative Robot-Manipulators in the Concepts of Industry 5.0* (Doctoral dissertation, Collection of scientific papers «SCIENTIA»).

Науковий керівник: Демська Наталія Павлівна, доцент, кандидат технічних наук, доцент кафедри КІТАР Харківського національного університету радіоелектроніки

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2025

(Випуск 1)

[електронне видання]

Харків 2025

АНАЛІЗ МЕТОДІВ ОПТИМІЗАЦІЇ РОБОТИ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ПРИ НАВАНТАЖЕННІ

Обривко Є.В.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: yevhen.obryvko@nure.ua

В роботі розглянуто актуальні питання оптимізації роботи систем дистанційного навчання при значних навантаженнях. Проаналізовано існуючі методи оптимізації, їх ефективність та особливості застосування, а також розглянуті підходи до визначення оптимальних параметрів конфігурації системи.

Ключові слова: оптимізація, дистанційне навчання, навантаження, стабільність, система.

ANALYSIS OF METHODS FOR OPTIMIZING THE OPERATION OF DISTANCE LEARNING SYSTEMS UNDER LOAD

Obryvko E. V.

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky Ave. 14

E-mail: yevhen.obryvko@nure.ua

The paper considers current issues of optimizing the operation of distance learning systems under significant loads. Existing optimization methods, their effectiveness and application features are analyzed, and approaches to determining optimal system configuration parameters are considered.

Key words: optimization, distance learning, load, stability, system.

АКТУАЛЬНІСТЬ РОБОТИ. В умовах стрімкого зростання популярності та поширення систем дистанційного навчання, питання забезпечення їхньої стабільної та ефективної роботи набуває першочергового значення. Сучасні освітні платформи часто обслуговують велику кількість користувачів одночасно, що створює значне навантаження на їхні ресурси.

Недостатня оптимізація системи може призвести до ряду негативних наслідків, включаючи зниження продуктивності та збільшення часу відгуку системи, що негативно впливає на досвід користування, перебої в роботі та недоступність платформи у періоди пікового навантаження, що може призвести до зриву навчального процесу, збільшення витрат на підтримку інфраструктури, необхідність масштабування ресурсів для забезпечення стабільної роботи.

Крім того, важливо враховувати постійний розвиток технологій та зміну вимог до подібних систем. З'являються нові функції, такі як відеоконференції, інтерактивні дошки, системи адаптивного навчання, які створюють додаткове навантаження на систему. Також зростають вимоги до безпеки та захисту даних, що передбачає впровадження додаткових механізмів, які також можуть впливати на продуктивність.

Тому, аналіз та розробка ефективних методів оптимізації роботи системи дистанційного навчання є критично важливим завданням. Це дозволить забезпечити високу доступність, продуктивність та масштабованість систем дистанційного навчання, що, в свою чергу, сприятиме підвищенню якості освіти, задоволеності користувачів та економічній ефективності навчальних закладів. Дослідження в цій області мають велике значення для створення надійних та зручних освітніх середовищ, які відповідають сучасним викликам та потребам.

ВСТУП. Для ефективної роботи систем дистанційного навчання в умовах високого навантаження, критично важливим є розуміння основних факторів, що впливають на їх

продуктивність, та потенційних проблем, які можуть виникнути. Основними проблемами є:

- збільшення часу відгуку – велика кількість одночасних користувачів може призвести до затримок у обробці запитів, що негативно впливає на зручність використання;
- перевантаження серверів – пікові навантаження можуть спричинити перевантаження серверів, що призводить до зниження продуктивності або навіть відмови в обслуговуванні;
- неефективне використання ресурсів – за рахунок поганої оптимізації доступні ресурси можуть неефективно використовуватись, тому такі фактори, як пропускна здатність мережі, процесорний час та пам'ять, можуть також негативно впливати на стабільність роботи системи.

ВПЛИВ НАВАНТАЖЕНЬ НА РОБОТУ СИСТЕМИ. В умовах дистанційного навчання системи часто стикаються зі значними коливаннями в кількості користувачів, що одночасно працюють з платформою. Це може бути пов'язано з піковими періодами активності, наприклад, під час здачі завдань, проведення онлайн-лекцій або тестувань. Такі стрибки навантаження можуть мати серйозний вплив на продуктивність та стабільність роботи системи.

Зі збільшенням кількості активних користувачів зростає кількість запитів до сервера, що обробляє систему дистанційного навчання. Це призводить до збільшення використання ресурсів сервера, таких як процесорний час, пам'ять та пропускна здатність мережі. Якщо система не оптимізована належним чином, це може призвести до збільшення часу відгуку (якщо користувачі відчувають затримки при завантаженні сторінок, відправці повідомлень або виконанні інших дій), зниження продуктивності (коли система починає працювати повільніше, що негативно впливає на навчальний процес) та перебоїв у роботі [1].

На рисунку 1 представлено графік впливу кількості користувачів на час відгуку системи.

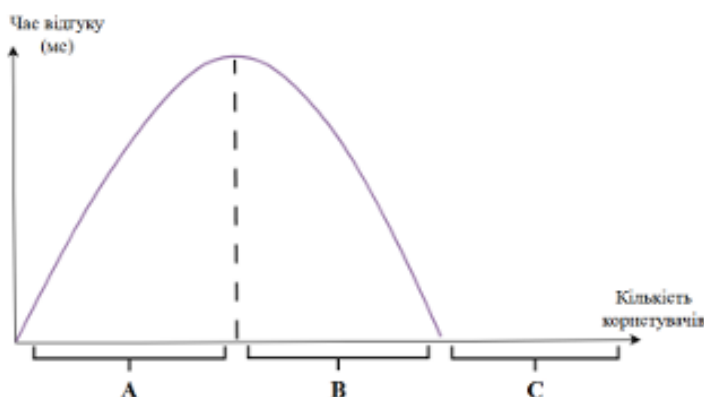


Рисунок 1 – Графік впливу кількості користувачів на час відгуку системи

На графіку (рис. 1) є кілька ділянок, що демонструють залежності, їх описано нижче:

Ділянка А – початкова ділянка, де кількість користувачів невелика. На цій ділянці час відгуку системи зростає відносно лінійно. Це означає, що система здатна ефективно обробляти запити, і кожен новий користувач трохи збільшує загальний час відгуку.

Ділянка В – перехідна ділянка, де кількість користувачів починає збільшуватися, наближаючись до меж можливостей системи. На цій ділянці час відгуку починає зростати швидше, і залежність стає нелінійною.

Ділянка С – ділянка перевантаження. Кількість користувачів перевищує можливості системи, і час відгуку різко зростає. Система може працювати нестабільно, спостерігаються

затримки, помилки або навіть відмови в обслуговуванні.

Як видно з графіка (рис. 1), час відгуку системи має тенденцію до збільшення зі зростанням кількості користувачів. Спочатку це збільшення може бути незначним, але після певного порогу час відгуку починає зростати експоненціально, що свідчить про перевантаження системи.

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ. Для забезпечення ефективної роботи системи дистанційного навчання при високих навантаженнях, важливо розглянути та проаналізувати існуючі методи оптимізації. До таких методів належать:

- кешування даних. Тимчасове зберігання часто використовуваних даних для зменшення кількості запитів до бази даних та прискорення часу відгуку системи;
- балансування навантаження. Розподіл вхідних запитів між кількома серверами для запобігання перевантаженню одного сервера та забезпечення стабільної роботи системи;
- оптимізація бази даних. Налаштування параметрів бази даних, індексування таблиць та оптимізація запитів для підвищення швидкості обробки даних;
- використання CDN (Content Delivery Network). Розміщення статичного контенту (навчальні матеріали, зображення, відео) на мережі розподілених серверів для прискорення завантаження контенту для користувачів з різних географічних локацій;
- асинхронна обробка завдань. Виконання певних завдань (наприклад, генерація звітів, розсилка повідомлень) у фоновому режимі, щоб не блокувати основні операції системи.

У таблиці 1 представлено результати порівняльного аналізу вищезгаданих методів оптимізації роботи системи дистанційного навчання. При порівнянні було враховано такі параметри, як масштабування системи, вплив на продуктивність роботи, та особливості впровадження.

Таблиця 1 – Порівняння методів оптимізації роботи системи дистанційного навчання.

Метод оптимізації	Масштабування системи	Вплив на продуктивність	Особливості впровадження
Кешування даних	Високе	Помірний	Підходить для часто використовуваних даних.
Балансування навантаження	Високе	Високий	Передбачає необхідність налаштування додаткової інфраструктури. (може впливати на швидкість обробки даних)
Оптимізація бази даних	Високе	Високий	Потребує додаткових ресурсів та знань для адміністрування БД.
Використання CDN	Високе	Помірний	Залежить від обсягу контенту та географії користувачів.
Асинхронна обробка завдань	Високе	Низький	Підходить для завдань, які не потребують миттєвої обробки.

1. Кешування даних. Кешування – це процес тимчасового зберігання копій даних у місці, яке забезпечує швидший доступ до них, ніж отримання даних з їхнього початкового розташування. У системах дистанційного навчання це означає збереження часто використовуваних даних (наприклад, вміст курсів, сторінки профілів, результати тестів) у пам'яті сервера або на пристрої користувача.

Приклади застосування в системах дистанційного навчання: кешування вмісту курсів (зберігання HTML-сторінок, PDF-файлів, відеолекцій), (кешування даних користувачів: Зберігання інформації профілів, результатів тестів), кешування статичних ресурсів

(зображення, CSS-файли, JavaScript-файли) [2].

2. Балансування навантаження. Це розподіл вхідних мережевих запитів або навантаження додатків між кількома серверами. Це дозволяє жодному окремому серверу не бути перевантаженим, забезпечуючи доступність та швидкість реагування додатків.

Приклади застосування в системах дистанційного навчання: розподіл трафіку між веб-серверами: Обробка HTTP-запитів користувачів, балансування навантаження на сервери баз даних, розподіл запитів на читання та запис даних.

На рисунку 2 представлено модель реалізації балансування навантаження.



Рисунок 2 – Модель балансування навантаження в системі [3]

Компонентами такої моделі є: клієнти (користувачі), які надсилають запити до системи. Це можуть бути студенти, викладачі, адміністратори, які намагаються отримати доступ до навчальних матеріалів, здати завдання, переглянути оцінки тощо.

Балансувальник навантаження – центральний елемент схеми, задача якого полягає у розподілі вхідних запитів від клієнтів між доступними серверами:

- черги – вхідні запити від клієнтів спочатку потрапляють у черги. Кожна черга може відповідати певному типу запитів, пріоритету або іншим критеріям. Черги допомагають тимчасово зберігати запити та обробляти їх у порядку надходження;

- процедура обслуговування. Це алгоритм або набір правил, за якими балансувальник навантаження визначає, на який саме сервер направити кожен запит. Існують різні алгоритми балансування, такі як Round Robin (запити розподіляються по черзі між серверами), Least Connections (запит направляється на сервер з найменшою кількістю поточних з'єднань), IP Hash (запити від одного і того ж IP-адреси завжди направляються на один і той же сервер), Weighted (сервери можуть мати різний пріоритет, і запити розподіляються з його урахуванням);

- повернення. Після обробки запиту сервером, результат повертається через балансувальник навантаження до відповідного клієнта;

- потоки втрачених пакетів. Якщо запит не може бути оброблений, він може бути втрачений. Балансувальник може реєструвати такі випадки для подальшого аналізу та оптимізації;

- пакети, що були витіснені. У деяких випадках, коли черги переповнюються, балансувальник може "витіснити" (видалити) деякі запити, щоб звільнити місце для нових [3].

Сервери – фактично обробляють запити клієнтів. У системі дистанційного навчання це

20

можуть бути сервери, на яких зберігаються навчальні матеріали, бази даних з інформацією про студентів та викладачів, сервери додатків, що забезпечують функціональність платформи.

3. Оптимізація бази даних. Оптимізація бази даних – це набір технік, спрямованих на підвищення продуктивності та ефективності роботи системи управління базами даних (СУБД) [4].

Приклади застосування в системах дистанційного навчання: оптимізація запитів для отримання списку курсів, завдань, оцінок та індексування таблиць з даними користувачів, курсів, результатів тестів.

4. Використання CDN (Content Delivery Network). CDN – це географічно розподілена мережа проксі-серверів та їхніх центрів обробки даних. Мета CDN полягає у наданні контенту користувачам з високою доступністю та продуктивністю.

Коли користувач запитує певний контент (наприклад, зображення, відео, CSS, JavaScript), CDN перенаправляє запит на найближчий до користувача сервер, який має копію цього контенту. Це зменшує відстань, яку повинні пройти дані, і прискорює завантаження [5].

Приклади застосування в системах дистанційного навчання: розміщення зображень курсів, логотипів, відеолекцій на CDN, розміщення CSS-файлів та JavaScript-файлів інтерфейсу користувача на CDN.

5. Асинхронна обробка завдань. Асинхронна обробка завдань – це техніка, при якій виконання певних завдань відбувається у фоновому режимі, незалежно від основного потоку виконання програми. Це дозволяє підвищити швидкість реагування системи на запити користувачів.

Коли користувач ініціює завдання, яке потребує значного часу на виконання, система не виконує це завдання одразу. Замість цього вона додає завдання до черги, а потім одразу ж повертає відповідь користувачеві. Фоновий процес бере завдання з черги та виконує їх [6].

Приклади застосування в системах дистанційного навчання: генерація звітів про успішність студентів, розсилка сповіщень про нові завдання, терміни здачі, обробка завантажених користувачами файлів (наприклад, перетворення форматів).

ВИСНОВКИ. Для поставленої задачі, коли система створюється під конкретний навчальний заклад, тоді використання CDN можна відкинути, оскільки географія використання системи не є сильно розподіленою і більша частина користувачів географічно зосереджена в одному місці (приблизно). Асинхронна обробка завдань також не має великого сенсу для впровадження в даному контексті, оскільки має низький вплив на продуктивність та швидкодію системи.

А такі методи, як кешування даних, балансування навантаження та оптимізація бази даних найбільше підходять для виконання поставленої задачі оптимізації стабільності роботи системи дистанційного навчання, адже мають високий вплив на розподілення навантаження та пришвидшення роботи системи.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Технології розробки WEB-ресурсів [Електронний ресурс] : навчальний посібник / В. П. Молчанов, О. К. Пандорін. – Харків :ХНЕУ ім. С. Кузнеця, 2019. – 130 с.

2. Що таке кешування даних – визначення, види, типи та ін.: resit.com.ua. URL: [https://resit.com.ua/scho-take-keshirovanie-dannyh-viznachennya-vidy-tipy-i-dr/#:~:text=Це%20процес%20тимчасового%20зберігання%20даних,скорочення%20часу%20до%20до%20даних.%20\(дата%20звернення%20:21.04.2025\).](https://resit.com.ua/scho-take-keshirovanie-dannyh-viznachennya-vidy-tipy-i-dr/#:~:text=Це%20процес%20тимчасового%20зберігання%20даних,скорочення%20часу%20до%20до%20даних.%20(дата%20звернення%20:21.04.2025).)

3. Іванісенко І.М. Методи балансування навантаження у розподілених системах з урахуванням самоподібних властивостей вхідних потоків. ХНУРЕ. , Харків. 2017, с. 117 – 120.

4. Оптимізація бази даних: шлях до високої продуктивності.: whileweb.com. URL:

<https://whileweb.com/uk/blog/optimizaciya-bazi-danih-shlyah-do-visokoyi-produktivnosti/> (дата звернення: 22.04.2025).

5. Content delivery network: Вікіпедія. URL: https://en.wikipedia.org/wiki/Content_delivery_network (дата звернення: 22.04.2025).

6. Асинхронна обробка каскадних транзакцій. Microsoft Learn. URL: <https://learn.microsoft.com/uk-ua/power-platform/admin/async-cascading> (дата звернення: 23.04.2025).

7. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic user authentication key for access to HMI/SCADA via unsecured internet networks. *Computational intelligence and neuroscience*, 2022(1), 5866922.

8. Demska, N., Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2025). ANALYSIS OF METHODS, MODELS AND ALGORITHMS FOR A COLLABORATIVE ROBOTS GROUP DECENTRALIZED CONTROL. *ACUMEN: International journal of multidisciplinary research*, 2(2), 235-249.

9. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. *International Journal of Crowd Science*, 9(1), 29-43.

10. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224-239.

11. Yevsieiev V. Comparison of Functional Capabilities of Classic Manipulator Robots and Collaborative Robots / V. Yevsieiev, N. Demska // Digital innovation & sustainable development 2024 : Proceedings of I st International Conference, November 15, 2024. - Kharkiv, 2024. - P.16-17.

12. Maksymova, S., Yevsieiev, V., Abu-Jassar, A., & Al-Abdallat, M. (2023). Prosthetic Hand 3d Model Development. *Multidisciplinary Journal of Science and Technology*, 3(5), 147-156.

13. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). MATHEMATICAL MODEL BASED ON MULTI-AGENT REINFORCEMENT LEARNING (MARL) AND PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP) FOR MODELING CARGO MOVEMENT FOR A MOBILE ROBOTS GROUP. *Multidisciplinary Journal of Science and Technology*, 5(4), 480-489.

14. Attar, H., Abu-Jassar, A. T., Yevsieiev, V., Lyashenko, V., Nevliudov, I., & Luhach, A. K. (2022). Zoomorphic mobile robot development for vertical movement based on the geometrical family caterpillar. *Computational intelligence and neuroscience*, 2022(1), 3046116.

15. Attar, H., Abu-Jassar, A. T., Amer, A., Lyashenko, V., Yevsieiev, V., & Khosravi, M. R. (2022). Control system development and implementation of a CNC laser engraver for environmental use with remote imaging. *Computational intelligence and neuroscience*, 2022(1), 9140156.

16. Nevliudov, I., Yevsieiev, V., Maksymova, S., Demska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In *2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)* (pp. 61-64). IEEE.

17. Khalid, M. S., Yevsieiev, V., Nevliudov, I. S., Lyashenko, V., & Wahid, R. (2022). HMI development automation with GUI elements for object-Oriented programming Languages implementation.

Науковий керівник: Демська Наталія Павлівна, кандидат технічних наук, доцент кафедри КІТАР Харківського національного університету радіоелектроніки

ДОДАТОК В
Демонстраційний матеріал

