

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

Б. П. Бочаров, М. Ю. Воєводіна

РОЗРОБКА КЛІЄНТСЬКИХ ТА СЕРВЕРНИХ ВЕБДОДАТКІВ
(FRONT-END/BACK-END)

КОНСПЕКТ ЛЕКЦІЙ

*(для здобувачів 3 курсу першого (бакалаврського) рівня
вищої освіти денної форми навчання
зі спеціальностей 126, F6 – Інформаційні системи та технології)*

Харків
ХНУМГ ім. О. М. Бекетова
2025

Бочаров Б. П. Розробка клієнтських та серверних вебдодатків (FRONT-END/BACK-END) : конспект лекцій для здобувачів 3 курсу першого (бакалаврського) рівня вищої освіти денної форми навчання зі спеціальностей 126, F6 – Інформаційні системи та технології / Б. П. Бочаров, М. Ю. Воєводіна; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2025. – 68 с.

Автори:

канд. техн. наук, доц. Б. П. Бочаров,
ст. викл. М. Ю. Воєводіна

Рецензент

О. Б. Костенко, доцент кафедри комп'ютерних наук та інформаційних технологій Харківського національного університету міського господарства ім. О. М. Бекетова

Рекомендовано кафедрою комп'ютерних наук та інформаційних технологій, протокол № 17 від 2 травня 2025 р.

© Б. П. Бочаров, М. Ю. Воєводіна, 2025
© ХНУМГ ім. О. М. Бекетова, 2025

ЗМІСТ

ВСТУП	4
Підготовка інформації	6
Проектування бази даних.....	6
Створення бази даних за допомогою монітора MySQL	8
Виконання команд у пакетному режимі.....	12
Підготовка питань.....	16
Робота системи	22
Зв'язок із базою даних і аварійний вихід	24
Створення початкової сторінки.....	25
Реєстрація нового користувача	26
Запис у базу нового користувача	29
Створення особистої сторінки користувача.....	33
Створення нового тесту	42
Виконання тесту.....	46
Вхід адміністратора в систему	56
Сторінка адміністратора	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67

ВСТУП

Вебтехнології стрімко розвиваються, отримують все більше нових можливостей, стають більш зручними для користувачів. Саме тому уміння розробляти привабливі, функціональні, зручні вебдодатки наразі стають важливими складовими інформаційної культури фахівця у галузі інформаційних технологій.

Дисципліна «Розробка клієнтських та серверних вебдодатків (FRONT-END/BACK-END)» є частиною професійної підготовки фахівця в галузі інформаційних систем і технологій. Цикл лекцій спрямований на практичне вирішення професійної прикладної задачі від постановочної частини до кінцевого програмного результату.

Процес створення сучасного вебдодатку буде розглянуто детально, всебічно і на конкретних прикладах.

Як задачу, обрано створення системи тестування знань. Не варто забувати, що процес початку роботи зі створення будь-якого програмного продукту починається з вивчення предметної області. А саме ця предметна область наближена до діяльності студентської спільноти.

Вам необхідно буде виконати такі завдання:

1. Постановка завдання створення повноцінного вебдодатка:

- встановлення та налаштування програмного забезпечення WAMP (Windows – Apache – MariaDB / MySQL – PHP / Perl / Python) на локальному комп'ютері;
- розробка структури бази даних системи тестування знань;
- створення бази даних різними сервісними утилітами;
- наповнення бази даних початковою інформацією за допомогою SQL.

2. Розробка клієнтської частини завдання:

- ознайомлення з клієнтською частиною завдання;
- створення таймера за допомогою скрипта JavaScript;
- створення функції перевірки введення користувача за допомогою скрипта JavaScript.

3. Розробка серверної частини завдання:

- ознайомлення з серверною частиною завдання;
- створення функції зв'язку з базою даних;
- створення головного (початкового) файлу завдання;
- створення файлів реєстрації користувачів;
- створення файлів генерування нових тестів випадковим чином;

- реєстрація в Microsoft Azure і отримання підписки на створення і розгортання безкоштовних хмарних додатків, передбачених безкоштовною ліцензією ХНУМГ;
- створення та розгортання безкоштовних хмарних додатків Microsoft Azure;
- перенесення свого вебдодатку на платформу Microsoft Azure.

УВАГА! Посилання на рядки вихідного тексту здійснюються за допомогою знака «№» (номер).

Підготовка інформації

У цьому розділі ми розглянемо такі питання:

- проєктування бази даних – розроблення всіх таблиць, інформаційних зв'язків і полів;
- створення бази даних за допомогою монітора MySQL – обмін файлами з сервером, виклик програм на сервері, робота з монітором MySQL, створення з його допомогою таблиць бази;
- підготовка питань – зі спеціального файлу створюються тексти питань та набір відповідей. Неправильні відповіді беруться з сусідніх питань. Формуються SQL-запити і вставляються в базу.

Проектування бази даних

Нижче подана структура бази даних і опис всіх таблиць.

Інформація подається за допомогою таких таблиць:

- usr – користувачі;
- def – опис тестів;
- tst – тести;
- q – питання.

Інформаційні зв'язку (реляційні відносини) між таблицями подані на рисунку 1:

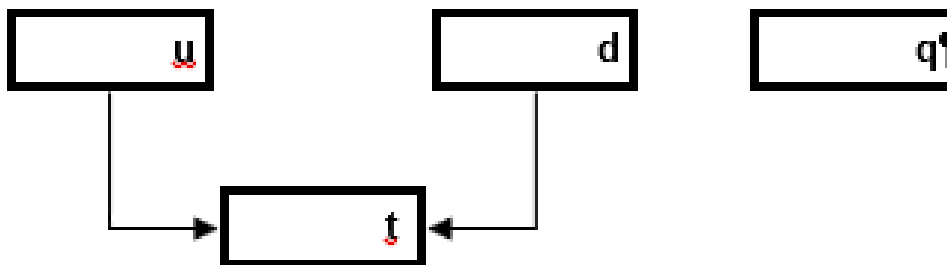


Рисунок 1 – Реляційні відносини між таблицями бази даних

Нижче наведено опис інформаційної структури всіх таблиць. Були використані такі терміни:

- auto_increment – цілочисельне поле, значення якого автоматично збільшується на 1 для кожного нового запису;
- int – цілочисельне поле;
- unsigned – невід’ємне цілочисельне поле;
- NULL – пусте поле;
- NOT NULL – непорожнє поле;
- varchar (N) – рядок символів довжиною N;
- default – значення за замовчуванням для поля (default " – порожній рядок);

- text – рядок символів довільної довжини;
 - PRIMARY KEY – головний індекс таблиці;
 - UNIQUE KEY – унікальний (значення для різних записів не збігаються)
- індекс;
- KEY – індекс.

Поля таблиці usr (опис поля в квадратних дужках):

- usr_id [int unsigned NOT NULL auto_increment] – номер користувача;
- usr_als [varchar (16) NOT NULL default ""] – псевдонім користувача;
- usr_psw [varchar (16) default NULL] – пароль користувача;
- usr_name [varchar (16) default NULL] – ім'я користувача;
- usr_mname [varchar (16) default NULL] – по батькові користувача;
- usr_lname [varchar (16) default NULL] – прізвище користувача;
- usr_add [text] – додаткова інформація про користувача.

Індекси таблиці usr:

- PRIMARY KEY (usr_id);
- UNIQUE KEY ind_als (usr_als).

Поля таблиці def:

- def_id [int (10) unsigned NOT NULL default 0] – номер опису;
- def_name [varchar (64) NOT NULL default ""] – назва тесту;
- def_rule [text] – правила побудови тесту;

Індекс таблиці def – PRIMARY KEY (def_id).

У поле def_rule (правила побудови тесту) записується така інформація: номер групи питань, пробіл, кількість питань з групи в тесті. Інформація вводиться для кожної групи (роздільник – символ #).

Поля таблиці tst:

- tst_id [int unsigned NOT NULL auto_increment] – номер тесту;
- usr_id [int unsigned NOT NULL default 0] – номер користувача;
- def_id [int unsigned NOT NULL default 0] – номер опису тесту;
- tst_q [text] – питання тесту;
- tst_a [text] – відповіді тесту;
- tst_ok [text] – правильні відповіді;
- tst_bal [text] – кількості балів за правильну відповідь;
- tst_tm [int NOT NULL default 0] – час, витрачений на проходження тесту.

Індекси таблиці tst:

- PRIMARY KEY (tst_id);
- KEY ind_usr (usr_id);
- KEY ind_def (def_id).

У поле tst_q (питання тесту) під час формування тесту вводяться номери питань, розділені пробілами.

У поле `tst_ok` (правильні відповіді) під час формування тесту вводяться номери правильних відповідей для кожного питання, розділені пробілами.

У поле `tst_bal` під час формування тесту вводяться кількості балів за правильну відповідь на кожні питання, розділені пробілами.

У поле `tst_a` (відповіді тесту) на етапі проходження тесту вводяться відповіді на кожне запитання, розділені пробілами.

У поле `tst_tm` заноситься час проходження тесту.

Поля таблиці `q`:

- `q_id` [int unsigned NOT NULL auto_increment] – номер питання;
- `q_gr` [int unsigned NOT NULL default 0] – номер групи питань;
- `q_tm` [int unsigned NOT NULL default 0] – час на відповідь;
- `q_q` [text NOT NULL] – текст питання;
- `q_a` [text NOT NULL] – текст відповідей;
- `q_ok` [int unsigned NOT NULL default 0] – номер правильної відповіді;
- `q_bal` [int unsigned NOT NULL default 0] – кількість балів за правильну відповідь.

Індекси таблиці `q`:

- PRIMARY KEY (`q_id`);
- KEY `ind_gr` (`q_gr`).

Створення бази даних за допомогою монітора MySQL

Ми будемо створювати базу даних безпосередньо на сервері, на якому встановлена UNIX-подібна операційна система. Для цього нам знадобляться:

1. SSH-клієнт (Secure SHell, «безпечна оболонка») – програма, що дозволяє виробляти віддалене управління операційною системою на сервері. Таких програм дуже багато, зазвичай провайдери поміщають таку програму на своєму сайті. Ми користуємося програмою PUTTY.

2. FTP-клієнт (File Transfer Protocol, протокол передачі файлів) – програма, призначена для обміну файлів із сервером. Ми використовуємо файловий менеджер FAR.

Всю інформацію, пов'язану з налаштуванням SSH і FTP доступу, можна отримати у провайдера або системного адміністратора.

Почнемо зі входу в систему. Після вказівки адреси сервера, SSH-клієнт просить вказати ім'я користувача і пароль.

На екрані можна побачити приблизно такий текст:

```
login as: username
Using keyboard-interactive authentication.
Password:
Last login: Mon May  2 08:50:29 2011 from 46.118.220.174
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
    The Regents of the University of California. All rights
reserved.
        Welcome!
[homedir ~]$
```

Всі використовувані далі імена та адреси умовні:

- **username** – ім'я користувача в операційній системі сервера;
- **homedir** – домашній каталог користувача на сервері;
- **dbhost** – хост (адреса) сервера MySQL;
- **dbuser** – користувач базою даних MySQL;
- **dbname** – ім'я бази даних MySQL.

Рядок `[homedir ~]$` – запрошення для введення команд.

Щоб керувати базою даних, використовується монітор MySQL, який запускається командою «mysql» у такий спосіб:

```
[homedir ~]$ mysql -h dbhost -u dbuser -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12132978
Server version: 5.0.90-log FreeBSD port: mysql-server-5.0.90
Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.
mysql>
```

Рядок `mysql>` – запрошення для введення команд монітора.

Наступна команда вибирає конкретну базу даних.

```
mysql> use dbname;
Database changed
mysql>
```

За допомогою команди «show tables» можна переглянути всі таблиці в базі.

```
mysql> show tables;
Empty set (0.00 sec)
mysql>
```

Створимо таблицю `usr`. Команда «CREATE TABLE» дуже довга, тому ми будемо вводити порядково. Рядок `->` – запрошення ввести продовження команди. У кінці кожного рядка натискаємо «ENTER», закінчення команди символ «;» (крапка з комою).

```
mysql> CREATE TABLE usr (
->   usr_id int unsigned NOT NULL auto_increment,
->   usr_als varchar(16) NOT NULL default '',
->   usr_psw varchar(16) default NULL,
->   usr_name varchar(16) default NULL,
->   usr_mname varchar(16) default NULL,
->   usr_lname varchar(16) default NULL,
->   usr_add text,
->   PRIMARY KEY (usr_id),
->   UNIQUE KEY ind_als (usr_als)
-> ) TYPE=MyISAM;
Query OK, 0 rows affected, 1 warning (0.01 sec)
mysql>
```

Ще раз переглянемо всі таблиці в базі.

```
mysql> show tables;
+-----+
| Tables_in_dbname |
+-----+
| usr               |
+-----+
1 row in set (0.00 sec)
mysql>
```

Тепер ми бачимо таблицю `usr`. Видалити таблицю можна за допомогою команди «`DROP TABLE IF EXISTS`». Виконаємо її і переконаємося, що таблиця дійсно вилучена.

```
mysql> DROP TABLE IF EXISTS usr;
Query OK, 0 rows affected (0.00 sec)
mysql> show tables;
Empty set (0.00 sec)
mysql>
```

Створимо ще раз таблицю `usr` (текст команди див. вище) та інші таблиці: `def`, `tst`, `q`. Після цього знову переглянемо всі таблиці в базі.

```
mysql> CREATE TABLE def (
->   def_id int(10) unsigned NOT NULL default 0,
->   def_name varchar(64) NOT NULL default "",
->   def_rule text,
->   PRIMARY KEY (def_id)
-> ) TYPE=MyISAM;
Query OK, 0 rows affected, 1 warning (0.01 sec)
mysql> CREATE TABLE tst (
->   tst_id int unsigned NOT NULL auto_increment,
usr_id int unsigned NOT NULL default 0,
->   usr_id int unsigned NOT NULL default 0,
->   def_id int unsigned NOT NULL default 0,
->   tst_q text,
->   tst_a text,
->   tst_ok text,
->   tst_bal text,
->   tst_tm int NOT NULL default 0,
->   PRIMARY KEY (tst_id),
->   KEY ind_usr (usr_id),
->   KEY ind_def (def_id)
-> ) TYPE=MyISAM;
Query OK, 0 rows affected, 1 warning (0.02 sec)
mysql> CREATE TABLE q (
->   q_id int unsigned NOT NULL auto_increment,
->   q_gr int unsigned NOT NULL default 0,
->   q_tm int unsigned NOT NULL default 0,
->   q_q text NOT NULL,
->   q_a text NOT NULL,
->   q_ok int unsigned NOT NULL default 0,
->   q_bal int unsigned NOT NULL default 0,
```

```

-> PRIMARY KEY (q_id),
-> KEY ind_gr (q_gr)
-> ) TYPE=MyISAM;
Query OK, 0 rows affected, 1 warning (0.03 sec)
mysql> show tables;
+-----+
| Tables_in_dbname |
+-----+
| def                |
| q                  |
| tst                |
| usr                |
+-----+
4 rows in set (0.00 sec)
mysql>

```

Подивитися структуру таблиці можна за допомогою команди «describe». Подивимося структуру таблиці `usr`.

```

mysql> describe usr;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| usr_id     | int(10) unsigned   | NO   | PRI | NULL    | auto_in|
| usr_als    | varchar(16)        | NO   | UNI |         | crement|
| usr_psw    | varchar(16)        | YES  |     | NULL    |        |
| usr_name   | varchar(16)        | YES  |     | NULL    |        |
| usr_mname  | varchar(16)        | YES  |     | NULL    |        |
| usr_lname  | varchar(16)        | YES  |     | NULL    |        |
| usr_add    | text               | YES  |     | NULL    |        |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
mysql>

```

Подивимося структуру інших таблиць (`def`, `tst`, `q`).

```

mysql> describe def;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| def_id     | int(10) unsigned   | NO   | PRI | 0        |        |
| def_name   | varchar(64)        | NO   |     |          |        |
| def_rule   | text               | YES  |     | NULL    |        |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
mysql> describe tst;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| tst_id     | int(10) unsigned   | NO   | PRI | NULL    | auto_in|
| usr_id     | int(10) unsigned   | NO   | MUL | 0        | crement|
| def_id     | int(10) unsigned   | NO   | MUL | 0        |        |
| tst_q      | text               | YES  |     | NULL    |        |
+-----+-----+-----+-----+-----+-----+

```

```

| tst_a   | text          | YES |      | NULL |      |
| tst_ok  | text          | YES |      | NULL |      |
| tst_bal | text          | YES |      | NULL |      |
| tst_tm  | int(11)       | NO  |      | 0    |      |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
mysql> describe q;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| q_id  | int(10) unsigned | NO   | PRI | NULL    | auto_incr |
| q_gr  | int(10) unsigned | NO   | MUL | 0        |            |
| q_tm  | int(10) unsigned | NO   |      | 0        |            |
| q_q   | text          | NO   |      | NULL     |            |
| q_a   | text          | NO   |      | NULL     |            |
| q_ok  | int(10) unsigned | NO   |      | 0        |            |
| q_bal | int(10) unsigned | NO   |      | 0        |            |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
mysql>

```

Виконання команд у пакетному режимі

Перше знайомство з монітором MySQL можна вважати закінченим. Залишилося тільки навчитися виконувати команди в пакетному режимі. Це вимагає знання деяких команд операційної системи сервера (ми вважаємо, що на сервері використовується UNIX-подібна операційна система).

Вийдемо з монітора (можна набрати Ctrl + d) і повернемося в систему. На екрані ми знову побачимо рядок-запрошення для введення команд.

Створимо каталог для наших файлів. У принципі всі операції роботи з файлами можна виконувати за допомогою FTP-клієнта (працювати з програмою FAR значно зручніше). Однак, для загального розвитку, деякі операції з файлами будемо виконувати в операційній системі.

Каталог можна створити за допомогою команди «mkdir», а перейти в нього можна за допомогою команди «cd».

```

[homedir ~]$ mkdir sql_tst
[homedir ~]$ cd sql_tst
[homedir ~/sql_tst]$

```

Створено каталог sql_tst, його вміст можна переглянути за допомогою команди «ls -l» («ls» – команда, «-l» – опція).

```

[homedir ~/sql_tst]$ ls -l
[homedir ~/sql_tst]$

```

Оскільки каталог порожній, то команда перегляду нічого не показує. Створимо в FAR файл gol.bat, за допомогою якого ми викличемо монітор MySQL і змусимо його виконувати команди автоматично (у пакетному режимі).

```
mysql -h dbhost -u dbuser -p <q1.sql> q1_res.txt
```

Монітор виконує команди з файлу q1.sql, результат (інформація, що виводиться монітором) записується у файл q1_res.txt.

Тепер створимо файл q1.sql і запишемо в нього все команди створення таблиць бази даних.

```
USE dbname;
DROP TABLE IF EXISTS usr;
CREATE TABLE usr (
    usr_id int unsigned NOT NULL auto_increment,
    usr_als varchar(16) NOT NULL default '',
    usr_psw varchar(16) default NULL,
    usr_name varchar(16) default NULL,
    usr_mname varchar(16) default NULL,
    usr_lname varchar(16) default NULL,
    usr_add text,
    PRIMARY KEY (usr_id),
    UNIQUE KEY ind_als (usr_als)
) TYPE=MyISAM;
DROP TABLE IF EXISTS def;
CREATE TABLE def (
    def_id int(10) unsigned NOT NULL default 0,
    def_name varchar(64) NOT NULL default "",
    def_rule text,
    PRIMARY KEY (def_id)
) TYPE=MyISAM;
DROP TABLE IF EXISTS tst;
CREATE TABLE tst (
    tst_id int unsigned NOT NULL auto_increment,
    usr_id int unsigned NOT NULL default 0,
    def_id int unsigned NOT NULL default 0,
    tst_q text,
    tst_a text,
    tst_ok text,
    tst_bal text,
    tst_tm int NOT NULL default 0,
    PRIMARY KEY (tst_id),
    KEY ind_usr (usr_id),
    KEY ind_def (def_id)
) TYPE=MyISAM;
DROP TABLE IF EXISTS q;
CREATE TABLE q (
    q_id int unsigned NOT NULL auto_increment,
    q_gr int unsigned NOT NULL default 0,
    q_tm int unsigned NOT NULL default 0,
    q_q text NOT NULL,
    q_a text NOT NULL,
    q_ok int unsigned NOT NULL default 0,
    q_bal int unsigned NOT NULL default 0,
    PRIMARY KEY (q_id),
    KEY ind_gr (q_gr)
) TYPE=MyISAM;
```

Тепер необхідно відправити файлу по FTP на сервер в каталог sql_tst. Команда перегляду каталогу тепер показує таку інформацію.

```
[homedir ~/sql_tst]$ ls -l
-rw-r--r-- 1 username username 74 May 2 09:20 go1.bat
-rw-r--r-- 1 username username 1309 May 2 09:20 q1.sql
[homedir ~/sql_tst]$
```

Опис кожного файлу містить таку інформацію.

Права доступу до файлу (-rw-r - r--).

Число посилань до файлу (1). В операційній системі UNIX один і той же файл може знаходитися в різних каталогах під різними іменами. Швидше за все, ця інформація у цьому випадку не становить практичного інтересу.

Ім'я автора (власника) файлу (username).

Група, до якої належить власник файлу (username).

У цьому випадку ці імена збігаються.

Розмір файлу (74).

Дата створення (May 2 9:20).

Файл (go.bat).

Зауважимо, що файлова система UNIX розрізняє три категорії користувачів:

1. Власники файлів.
2. Члени групи, до якої належить власник файлу.
3. Всі інші користувачі.

Для кожної групи система розрізняє три типи доступу до файлу:

r – читання з файлу,

w – запис у файл,

x – виконання файлу.

В описі файлу (-rw-r - r--) перший символ (-) – це тип файлу (файл або каталог), наступні три (rw-) – права власника файлу (у цьому випадку він має право тільки читати і писати, але не виконувати файл). Наступні дві групи по три символи (r - r--) – права членів групи та інших користувачів (тільки читання).

Таким чином, у системі UNIX мало просто дати файлу розширення bat, щоб він виконувався як командний. Необхідно встановити для файлу відповідні права доступу. Це можна зробити за допомогою команди «chmod + x» (зауважимо, що команда «chmod -x» забороняє виконання файлу).

```
[homedir ~/sql_tst]$ chmod +x go.bat
[homedir ~/sql_tst]$ ls -l
-rwxr-xr-x 1 username username 74 May 2 09:20 go1.bat
-rw-r--r-- 1 username username 1309 May 2 09:20 q1.sql
[homedir ~/sql_tst]$
```

Тепер файл go.bat став виконуваним. Після виклику його буде обробляти спеціальний інтерпретатор команд shell. І остання особливість виклику

командних файлів у UNIX – для виклику файлу з поточного каталогу необхідно перед його ім'ям вказати набір символів «./» (точка і коса риска).

Ми так детально описуємо виклик програм і виконання команд у системі UNIX, тому це практично єдине, що не може зробити FAR (він не є клієнтом SSH). Всі інші файлові операції (навіть зміна прав доступу) можна, і значно зручніше виконувати в FAR.

Виконаємо командний файл go.bat (створення таблиць бази даних).

```
[homedir ~/sql_tst]$ ./go1.bat
Enter password:
[homedir ~/sql_tst]
```

У нас з'явився новий файл q1_res.txt. Розмір цього файлу – 0, це означає монітор MySQL не видав ніяких повідомлень. Таблиці створені успішно.

Створимо ще один файл, який дасть команді монітора видати інформацію про таблиці в базі і структурах кожної таблиці. Назвемо його q2.sql.

```
USE dbname;
SHOW TABLES;
DESCRIBE usr;
DESCRIBE def;
DESCRIBE tst;
DESCRIBE q;
```

Командний файл назвемо go2.sql.

```
mysql -h dbhost -u dbuser -p <q2.sql >q2_res.txt
```

Виконаємо цей файл, результат можна побачити у файлі q2_res.txt.

Tables_in_dbname

```
def
q
tst
usr
```

Field	Type	Null	Key	Default	Extra
usr_id	int(10) unsigned	NO	PRI	NULL	auto_increment
usr_als	varchar(16)	NO	UNI		
usr_psw	varchar(16)	YES		NULL	
usr_name	varchar(16)	YES		NULL	
usr_mname	varchar(16)	YES		NULL	
usr_lname	varchar(16)	YES		NULL	
usr_add	text	YES		NULL	

Field	Type	Null	Key	Default	Extra
def_id	int(10) unsigned	NO	PRI	0	
def_name	varchar(64)	NO			
def_rule	text	YES		NULL	

Field	Type	Null	Key	Default	Extra
tst_id	int(10) unsigned	NO	PRI	NULL	auto_increment
usr_id	int(10) unsigned	NO	MUL	0	
def_id	int(10) unsigned	NO	MUL	0	
tst_q	text	YES		NULL	
tst_a	text	YES		NULL	
tst_ok	text	YES		NULL	
tst_bal	text	YES		NULL	
tst_tm	int(11)	NO		0	

Field	Type	Null	Key	Default	Extra
q_id	int(10) unsigned	NO	PRI	NULL	auto_increment
q_gr	int(10) unsigned	NO	MUL	0	
q_tm	int(10) unsigned	NO		0	
q_q	text	NO		NULL	
q_a	text	NO		NULL	
q_ok	int(10) unsigned	NO		0	
q_bal	int(10) unsigned	NO		0	

Підготовка питань

Для попередньої підготовки питань використовують такі файли.

- imgw – каталог, у якому знаходяться всі зображення;
- w_1_2.txt – вихідні дані для формування питань;
- w1.awk – програма формування 1-ї групи питань;
- w2.awk – програма формування 2-ї групи питань;
- go1.bat – командний файл, який виконує всі програми;
- wg01.txt – результат роботи програми w1.awk;
- wg02.txt – результат роботи програми w2.awk.

Вихідні дані для питань подані в файлі w_1_2.txt. У непарних рядках вказані зображення іконок (елементів управління), а в парних – їхній опис.

```

1  wr0101.jpg
2  створити новий файл
3  wr0102.jpg
4  відкрити файл
5  wr0103.jpg
6  зберегти файл на жорсткому диску
7  wr0201.jpg
8  надрукувати файл
9  wr0202.jpg

```

Програма w1.awk формує текст питання і набір відповідей. Для питань використовуються імена файлів картинок, а для відповідей – опису елементів управління. Неправильні відповіді беруться з сусідніх питань. Правильна відповідь вказується першою.

Посилання на рядки вихідного тексту здійснюються за допомогою значка «№» (номер).

На початку роботи програми (№ 1–4) у вихідний файл записується загальна для всіх питань інформація:

1. Номер категорії (групи питань), у цьому випадку – 1.
2. Час, відведений на відповідь, – 30 (секунд).
3. Кількість балів за правильну відповідь – 1.

```
1 BEGIN{
2     printf("1\n30\n1\n*****\n");
3     n = 0;
4 }
```

Потім зчитується вихідна інформація для формування питань (№ 6–11). У масив arr1 записуються імена файлів картинок, а в масив arr2 – опису елементів управління. Кількість питань записується в змінну n.

```
6 {
7     arr1[n] = $0;
8     getline;
9     arr2[n] = $0;
10    n++;
11 }
```

Після читання вихідного файлу відбувається формування питань. Для кожного питання передбачено п'ять відповідей (з яких одна правильна). Змінні описують інтервал у масиві вихідної інформації, який використовується для генерації неправильних відповідей (№ 14–24).

```
13 END{
14     for(i=0; i<n; i++){
15         n1 = i-2;
16         n2 = i+2;
17         if(n1 < 0){
18             n1 = 0;
19             n2 = 4;
20         }
21         if(n2 >= n){
22             n2 = n-1;
23             n1 = n2-4;
24         }
25     }
```

У № 25 виводиться текст питання. Варто зазначити, що програма awk не сприймає символ «я» в початковому тексті. Цей символ представлений шістнадцятковим кодом \xFF. Потім виводиться правильну відповідь (№ 26) і неправильні відповіді (цикл в № 27–30).

```
25     printf("<img src='imgw/%s' align=left> Що можна
зробити за допомогою цього елемента керування?\xFF?\n",arr1[i]);
26     printf("# %s\n",arr2[i]);
27     for(j=n1; j<=n2; j++)
```

```

28         if(j != i)
29             printf("# %s\n",arr2[j]);
30     print "*****";
31 }
32 }

```

У результаті роботи програми виходить файл wg01.txt. Фрагмент цього файлу наведено нижче.

```

1      1
2      30
3      1
4      *****
5      <img src='imgw/wp0101.jpg' align=left> Що можна зробити
за допомогою цього елемента керування?
6      # створити новий файл
7      # відкрити файл
8      # зберегти файл на жорсткому диску
9      # надрукувати файл
10     # переглянути інформацію, що виводиться до друку
11     *****
12     <img src='imgw/wp0102.jpg' align=left> Що можна зробити
за допомогою цього елемента керування?
13     # відкрити файл
14     # створити новий файл
15     # зберегти файл на жорсткому диску
16     # надрукувати файл
17     # переглянути інформацію, що виводиться до друку
18     *****

```

Програма w2.awk працює аналогічно програмі w1.awk, за винятком того, що для питань використовуються опис елементів управління, а для відповідей – імена файлів картинок.

Номер категорії (групи питань), час, відведений на відповідь, і кількість балів за правильну відповідь записуються в № 2.

```

2      printf("2\n30\n1\n*****\n");

```

Текст питання та відповідь формуються в № 25–31.

```

25     printf("За допомогою якого елемента керування \xFF
можна %s?\n",arr2[i]);
26     printf("# <img src='imgw/%s'>\n",arr1[i]);
27     for(j=n1; j<=n2; j++)
28         if(j != i)
29             printf("# <img src='imgw/%s'>\n",arr1[j]);
30     print "*****";
31 }

```

У результаті роботи програми виходить файл wg02.txt. Фрагмент цього файлу наведено нижче.

```
1      2
2      30
3      1
4      *****
5      За допомогою якого елемента керування можна створити
новий файл?
6      # <img src='imgw/wp0101.jpg'>
7      # <img src='imgw/wp0102.jpg'>
8      # <img src='imgw/wp0103.jpg'>
9      # <img src='imgw/wp0201.jpg'>
10     # <img src='imgw/wp0202.jpg'>
11     *****
12     За допомогою якого елемента керування можна відкрити
файл?
13     # <img src='imgw/wp0102.jpg'>
14     # <img src='imgw/wp0101.jpg'>
15     # <img src='imgw/wp0103.jpg'>
16     # <img src='imgw/wp0201.jpg'>
17     # <img src='imgw/wp0202.jpg'>
18     *****
```

Виконання всіх програм здійснюється за допомогою командного файлу go1.bat.

```
1      ..\gawk --re-interval -f w1.awk w_1_2.txt >wg01.txt
2      ..\gawk --re-interval -f w2.awk w_1_2.txt >wg02.txt
```

Тепер запишемо питання в базу даних.

Файли, які використовуються для формування SQL-запитів:

- wg01.txt, wg02.txt – вихідні дані (отримані нижче);
- mk.awk – програма формування SQL-запитів, для кожної групи питань;
- un.awk – об'єднання SQL-запитів в один файл;
- go2.bat – командний файл, який виконує всі програми;
- qwe.txt – результат роботи програм.

Програма mk.awk формує SQL-запити з файлів wg01.txt і wg02.txt.

На початку тексту програми знаходяться допоміжні функції.

Функція «trim» видаляє з рядка символів «зайві» проміжки. Видаляються проміжок на початку і кінці рядка, будь-який підрядок, що складається з двох і більше проміжків замінюється на один пробіл.

```
1      function trim(str){
2          trim_str = str;
3          gsub(/^ +/, "", trim_str);
4          gsub(/ +$/, "", trim_str);
5          gsub(/ {2,}/, " ", trim_str);
```

```

6      return trim_str;
7  }

```

Функція «randint» повертає ціле випадкове число в інтервалі від 1 до n.

```

9  function randint(n){
10      return 1+int(n*rand());
11  }

```

Функція «out_q» (№ 13–37) виконує основну роботу програми – формування і висновок SQL-запитів. Функція приймає такі параметри:

- gr – номер групи (категорії) питань;
- tm – час на відповідь;
- q – текст питання;
- a – масив відповідей;
- n – кількість відповідей;
- bl – кількість балів за правильну відповідь.

```

13  function out_q(gr,tm,q,a,n,bl){

```

На початку роботи (№ 14–28) відповіді «перемішуються», тобто випадково формується масив ind, у який записуються номери відповідей. Номер правильної відповіді зберігається у змінній corr.

```

14      ind[1] = randint(n);
15      if(ind[1] == 1 ) corr = 1;
16      for(i=2; i<=n; i++){
17          pri = 0;
18          while (pri == 0){
19              ind[i] = randint(n);
20              pri = 1;
21              for(j=1; j<i; j++){
22                  if(ind[j] == ind[i]){
23                      pri = 0;
24                      break;
25                  }
26              }
27              if(ind[i] == 1 ) corr = i;
28          }

```

Потім у циклі (№ 30–35) з тексту відповіді видаляється службова інформація (№ 31–32). Всі відповіді записуються в одну змінну s (№ 33–34). Відповіді відокремлюються одна від одної символами ##.

```

29      s = "";
30      for(i=1; i<=n; i++){
31          s1 = a[ind[i]];
32          sub(/^#/,"",s1);
33          if(i != 1) s = s "##";
34          s = s trim(s1);
35      }

```

Виведення інформації здійснюється в № 36.

```

36     printf("INSERT INTO q VALUES (NULL, %d, %d, \"%s\",
\"%s\", %d, %d);\n",gr,tm,q,s,corr,bl);
37 }

```

На початку роботи програми читається загальна інформація (№ 40–43) і текст першого питання (№ 44).

```

39     BEGIN{
40         getline tstGr;
41         getline tstTm;
42         getline tstBl;
43         getline;
44         getline tstQ;
45         n=0;
46     }

```

Якщо прочитаний набір символів ***** (кінець питання), то здійснюється висновок відповідного SQL-запиту (№ 50), видаляється масив відповідей (№ 51) і читається текст наступного питання (№ 52). Потім обнуляється кількість відповідей (№ 53) і відбувається перехід до наступного запису (№ 54). В іншому випадку читається текст чергової відповіді (№ 56).

```

48     {
49         if($0 ~ /^\\*\\*\\*\\*\\*\\*/) {
50             out_q(tstGr,tstTm,tstQ,a,n,tstBl);
51             delete a;
52             getline tstQ;
53             n=0;
54             next;
55         }
56         a[++n] = $0;
57     }

```

Програма un.awk – об'єднання всіх SQL-запитів в один файл. На початку роботи виводиться ім'я бази даних (№ 10). Потім виводиться текст SQL-запиту (№ 14–15).

```

9      BEGIN{
10         print "USE dbname;";
11     }
12     {
13         s = trim($0);
14         if(s != "") print s;
15     }
16 }

```

Виконання всіх програм здійснюється за допомогою командного файлу go2.bat.

```
1    ..\gawk --re-interval -f mk.awk wg01.txt >w01.zzz
2    ..\gawk --re-interval -f mk.awk wg02.txt >w02.zzz
3    ..\gawk --re-interval -f un.awk w01.zzz w02.zzz >qwe.txt
```

У результаті роботи всіх програм виходить файл qwe.txt. Фрагмент цього файлу подано нижче (по одному SQL-запиту для кожної категорії питань).

```
1    USE dbname;
2    INSERT INTO q VALUES (NULL, 1, 30, "<img
src='imgw/wp0101.jpg' align=left> Що можна зробити за допомогою
цього елемента керування?", "зберегти файл на жорсткому
диску##створити новий файл##переглянути інформацію, що виводиться до
друку##надрукувати файл##відкрити файл", 2, 1);
79   INSERT INTO q VALUES (NULL, 2, 30, "За допомогою якого
елементу керування можна задати колір шрифту?", "<img
src='imgw/wp1101.jpg'>##
<img src='imgw/wp1102.jpg'>##
<img src='imgw/wp1103.jpg'>##
<img src='imgw/wp1003.jpg'>##
<img src='imgw/wp1004.jpg'>", 3, 1);
```

Нам залишилося сформулювати повністю сформулювати файл SQL-запиту, що створює базу даних.

У каталозі sql_tst створюємо файл (mkdb.bat).

```
mysql -h dbhost -u dbuser -p <mkdb.sql
```

Копіюємо файл q1.sql у файл mkdb.sql і додаємо після команд створення таблиці usr двох користувачів – адміністратора і тестового користувача.

```
INSERT INTO usr VALUES (NULL, "admin", "adminpsw", "", "", "", "");
INSERT INTO usr VALUES (
NULL, "iv", "ivaschenko", "Іващенко", "Павло", "Дмитрович", "тестовий
користувач");
```

Додаємо два опису тесту.

```
INSERT INTO def VALUES ("1", "Microsoft Word", "1 4#2 4#3 1#4 1#5
1#6 1#7 1#8 1#9 1#10 1#11 1#12 1#13 1#14 1");
INSERT INTO def VALUES ("2", "Microsoft Excel", "15 2#16 2#17 1#18
1#19 1#20 1#21 1#22 1#23 1#24 1#25 1#26 1#27 1#28 1#29 1#30 1#31
1#32 1#33 1#34 1#35 1#36 1");
```

Потім додаємо питання з файлу qwe.txt.

Робота системи

Система тестування знань повинна функціонувати у такий спосіб.

1. Користувач входить у систему на головну сторінку (файл index.html).

2. Якщо користувач ще не має облікового запису в системі, то йому пропонується її створити (файл newuser.html). Після створення облікового запису (файл newuser.php) користувач повертається на головну сторінку (п. 1).

3. Якщо користувач вже має обліковий запис, то він вводить псевдонім і пароль.

4. Відбувається перехід до особистій сторінці користувача (файл logon.php). На особистій сторінці відображаються результати всіх тестів.

5. Якщо для користувача вже підготовлений тест, то він розпочинає його виконання (файл test.php).

6. Користувач відповідає на питання і натискає кнопку «підтвердити», інформація передається серверу. Якщо час на відповідь минув, передача інформації відбувається автоматично. Потім виводиться наступне питання (файл test.php).

7. Після закінчення тесту користувачеві пропонується повернутися на особисту сторінку (п. 4).

8. Якщо для користувача ще не підготовлений тест, то йому надається можливість викликати формування нового тесту. Тип тесту користувач може вибрати зі списку (файл logon.php).

9. Здійснюється підготовка тесту (файл newtest.php). Після цього користувач може розпочати його виконання (п. 5).

Адміністрування системи (у цьому випадку адміністрування – це тільки отримання інформації) здійснюється за допомогою файлів admin.html і admin.php. Файл admin.html – сторінка входу адміністратора в систему, він там вводить логін і пароль. Файл admin.php видає інформацію про всі тести всіх користувачів.

Перейдемо до детального опису всіх елементів системи.

Файли php ми будемо називати також серверними сценаріями, програмами та програмами-обробниками.

Список файлів:

- 1) cnt.php – здійснює зв'язок із базою даних;
- 2) index.html – головна сторінка;
- 3) newuser.html – реєстрації нового користувача;
- 4) newuser.php – внесення в базу даних нового користувача;
- 5) newuser_php.html – результат роботи програми newuser.php;
- 6) logon.php – відображення особистої сторінки користувача;
- 7) logon_php1.html – результат роботи програми logon.php в разі, коли користувач не здав жодного тесту і йому видається форма підготовки нового тесту;
- 8) logon_php2.html – результат роботи програми logon.php в разі, коли користувач не здав жодного тесту і для нього вже підготовлений новий тест;
- 9) logon_php3.html – результат роботи програми logon.php в разі, коли користувач вже здав хоча б один тест і йому видається форма підготовки нового тесту;
- 10) newtest.php – створення нового тесту;
- 11) newtest_php.html – результат роботи програми newtest.php;
- 12) test.php – виконання тесту;

- 13) test_php1.html – перше питання тесту (користувач починає виконувати тест);
- 14) test_php2.html – четверте питання тесту;
- 15) test_php_fin.html – результат роботи системи після завершення тесту;
- 16) admin.html – головна сторінка адміністратора;
- 17) admin.php – відображення особистої сторінки адміністратора;
- 18) admin_php.html – результат роботи програми admin.php для двох тестових користувачів.

Зв'язок із базою даних і аварійний вихід

Файл cnt.php здійснює зв'язок з базою даних і аварійний вихід помилково.

Нагадаємо, що програма php на html-сторінці починається з символів «<?» і закінчується символами «?>». Імена змінні php починаються з символу «\$». Тип даних змінної визначається її поточним значенням.

У № 2–5 описані глобальні повідомлення (у цьому випадку повідомлення про помилки).

```
1    <?
2    $msg_err_cnt          ="<h1      align='center'><font
color='#FF0000'>Помилка з'єднання з базою даних!!!</font></h1>";
3    $msg_err_call         ="<h1      align='center'><font
color='#FF0000'>Неправильний виклик!!!</font></h1>";
4    $msg_err_psw          ="<h1      align='center'><font
color='#FF0000'>Неправильний пароль!!!</font></h1>";
5    $msg_not_reg          ="<h1 align='center'>Новий користувач не
zareestrovаний.<br>Спробуйте ще раз.</h1>";
```

Функція «tst_exit» (№ 7–10) здійснює виведення повідомлення, переданого їй у параметрі «\$ msg» (№ 8, оператор echo) і завершує роботу програми (№ 9, оператор exit).

Нагадаємо, що опис функції в php аналогічний опису функції в awk.

```
7    function tst_exit($msg){
8        echo $msg;
9        exit;
10   }
```

Функція «tst_connect_db» (№ 12–20) здійснює з'єднання з базою даних. Змінний \$ hst присвоюється ім'я хоста, змінний \$ dbn – ім'я бази даних, а змінний \$ psw – пароль (№ 13–15, всі значення умовні).

З'єднання здійснює вбудована функція «php mysql_connect» (№ 16), яка повертає дескриптор зв'язку з базою (використовується при подальшій роботі з базою). У разі невдалого з'єднання повертається false. У № 17 перевіряється коректність з'єднання, якщо сталася помилка, то здійснюється висновок повідомлення про помилку і вихід.

Вбудована функція «mysql_select_db» (№ 18) здійснює вибір бази даних (їх може бути декілька).

Функція «tst_connect_db» повертає дескриптор встановленого з'єднання з базою даних (№ 19).

```
12 function tst_connect_db($msg) {
13     $hst = "dbhost";
14     $dbn = "dbname";
15     $psw = "12345";
16     $db = mysql_connect($hst,$dbn,$psw);
17     if(!$db) tst_exit($msg);
18     mysql_select_db($dbn,$db);
19     return $db;
20 }
```

Функція «tst_connect_db» (№ 12–20) здійснює з'єднання з базою даних. Змінної \$ hst присвоюється ім'я хоста, змінної \$ dbn – ім'я бази даних, а змінної \$ psw – пароль (№ 13–15, всі значення умовні).

З'єднання здійснює вбудована функція «php mysql_connect» (№ 16), яка повертає дескриптор зв'язку з базою (використовується при подальшій роботі з базою). У разі невдалого з'єднання повертається false. У № 17 перевіряється коректність з'єднання, якщо сталася помилка, то здійснюється виведення повідомлення про помилку і вихід.

Вбудована функція «mysql_select_db» (№ 18) здійснює вибір бази даних (їх може бути декілька).

Функція «tst_connect_db» повертає дескриптор встановленого з'єднання з базою даних (№ 19).

```
22 function tst_exec_sql($q,$db,$msg) {
23     $r = mysql_query($q,$db);
24     if(!$r) tst_exit($msg);
25     return $r;
26 }
27 ?>
```

Створення початкової сторінки

Робота з системою починається з виклику файлу index.html.

```
1 <html>
2 <head>
3 <title>Тестування знань</title>
4 <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
5 <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6 --></style>
7 </head>
8 <body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0">
9 <form name="logon" method="post" action="logon.php">
```

```

10      <table      width="400"      border="0"      cellpadding="5" align="center">
11          <tr align="center">
12              <td colspan="2">
13                  <b><i><font                                size="+2"
color="#0000FF">Реєстрація</font></i></b></td>
14              </tr>
15          <tr>
16              <td width="30%" align="right"><b>Псевдо</b></td>
17              <td width="50%">
18                  <input      type="text"      name="als"      size="16"
maxlength="16">
19              </td>
20          </tr>
21          <tr>
22              <td width="30%" align="right"><b>Пароль</b></td>
23              <td width="50%">
24                  <input      type="password"      name="psw"      size="16"
maxlength="16">
25              </td>
26          </tr>
27          <tr align="center">
28              <td colspan="2">
29                  <input type="submit" value="Зареєструватися">
30              </td>
31          </tr>
32      </table>
33  </form>
34  <hr>
35  <h2 align="center"><a href="newuser.html">Новий
користувач</a></h2>
36  </body>
37  </html>

```

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

Форма реєстрації користувача (№ 9–33) називається «login», дані форми передаються програмі-обробнику (файлу) login.php (№ 9). Ім'я користувача заноситься в поле форми als (№ 18), а пароль – у полі psw (№ 24).

Якщо користувач ще не зареєстрований, то йому необхідно перейти за посиланням «новий користувач» (№ 35). У цьому випадку викликається файл newuser.html.

Реєстрація нового користувача

Файл newuser.html використовується для реєстрації нового користувача. № 1–6 і № 26–27 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

Для перевірки даних, введених користувачем, використовується функція «new_user_check», реалізована на мові JavaScript (№ 7–25).

У № 9–13 перевіряється правильність введення пароля (повинен бути не менше 6 символів і не більше 16).

Потім (№ 14–22) перевіряється заповнення інших полів форми (всі поля мають бути заповнені).

Якщо при перевірці виявлені помилки, то виводиться повідомлення (№ 11 або № 20) і користувач заповнює форму заново. Якщо все в порядку, то дані відправляються на сервер (№ 23).

```
1    <html>
2    <head>
3    <title>Реєстрація нового користувача</title>
4    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
5    <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
7    <script language="JavaScript">
8    function new_user_check(){
9        s_psw = new String(document.newuser.psw.value);
10       if(s_psw.length < 6){
11           window.alert("Занадто короткий пароль (має бути не
менше 6 символів и не більше 16)");
12       }
13       return;
14       s_al = new String(document.newuser.als.value);
15       s_nn = new String(document.newuser.name.value);
16       s_mn= new String(document.newuser.mname.value);
17       s_ln = new String(document.newuser.lname.value);
18       s_a = new String(document.newuser.add.value);
19       if((s_al.length == 0) || (s_nn.length == 0) ||
(s_mn.length == 0) || (s_ln.length == 0) || (s_a.length == 0)){
20           window.alert("Всі поля форми треба заповнити!!!");
21       }
22       return;
23       document.newuser.submit();
24   }
25   </script>
26   </head>
27   <body bgcolor="#FFFFFF" text="#000000">
```

Форма реєстрації нового користувача (рядки 36–82) називається «newuser», дані форми передаються програмі-обробнику (файлу) newuser.php (№ 36).

Ім'я користувача заноситься в поле форми als (№ 41), а пароль – у поле psw (№ 47).

Прізвище користувач вказує в поле форми lname (№ 53), ім'я – у поле name (№ 59), а по батькові – у поле mname (№ 65).

Додаткова інформація заноситься в поле форми add (№ 73).

Інформація відправляється на сервер за допомогою натискання кнопки «Зареєструватися» (№ 78).

До події onClick кнопки прив'язана функція JavaScript «new_user_check» (див. вище), яка перевіряє інформацію і остаточно відправляє її на сервер.

```
28     <table      width="100%"      border="0"      cellpadding="5"
cellpadding="0">
29         <tr>
30             <td      align="center"><font      size="+2"><b><i><font
color="#0000FF">Реєстрація          нового          користувача
</font></i></b></font></td>
31         </tr>
32         <tr>
33             <td align="center"><b><font color="#0000FF">всі поля
в формі мають бути заповнені</font></b></td>
34         </tr>
35     </table>
36     <form name="newuser" method="post" action="newuser.php">
37         <table      width="400"      border="0"      cellpadding="5" align="center">
38             <tr>
39                 <td width="102"><b>Псевдо</b></td>
40                 <td width="278">
41                     <input type="text" name="als" maxlength="16">
42                 </td>
43             </tr>
44             <tr>
45                 <td width="102"><b>Пароль</b></td>
46                 <td width="278">
47                     <input      type="password"      name="psw"
maxlength="16">
48                 </td>
49             </tr>
50             <tr>
51                 <td width="102"><b>Прізвище</b></td>
52                 <td width="278">
53                     <input type="text" name="lname">
54                 </td>
55             </tr>
56             <tr>
57                 <td width="102"><b>Ім'я</b></td>
58                 <td width="278">
59                     <input type="text" name="name">
60                 </td>
61             </tr>
62         </table>
```

```

63         <td width="102"><b>Отчество</b></td>
64         <td width="278">
65             <input type="text" name="mname">
66         </td>
67     </tr>
68     <tr align="center">
69         <td colspan="2"><b>Додаткова інформація</b></td>
70     </tr>
71     <tr align="center">
72         <td colspan="2">
73             <textarea                name="add"                cols="50"
rows="5"></textarea>
74         </td>
75     </tr>
76     <tr align="center">
77         <td colspan="2">
78             <input    type="button"    value="Зареєструватися"
onClick="new_user_check()" name="btnReg">
79         </td>
80     </tr>
81 </table>
82 </form>
83 </body>
84 </html>

```

Запис у базу нового користувача

Для внесення в базу даних нового користувача використовується файл (програма на мові php) newuser.php.

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

```

1     <html>
2     <head>
3     <title>Новий користувач</title>
4     <meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1251">
5     <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6     --></style>
7     </head>
8     <body bgcolor="#FFFFFF" text="#000000">

```

У № 10 вказується, що буде використаний файл cnt.php (див вище).

Потім перевіряються вхідні дані програми (№ 11). Всі змінні, що викликають із форми, передаються в програму php з тими ж іменами. Якщо не всі поля заповнені, то відбувається вихід із програми і виведення повідомлення про помилку (№ 12). Нагадаємо, що конкатенація рядків (використовується при

формуванні тексту повідомлення про помилку) у php здійснюється за допомогою оператора «.» (Крапка).

З'єднання з базою здійснюється в № 13.

Потім перевіряється наявність у базі користувача на псевдо \$ als. Текст запити формується в № 14, виконання запити – у № 15. Вбудована функція «php mysql_num_rows» показує кількість рядків, що повертаються запитом. Якщо повернуто ненульове кількість рядків, то користувач вже існує і відбувається аварійний вихід (№ 16–17).

Результат попереднього запити видаляється з пам'яті за допомогою вбудованої функції «php mysql_free_result» (№ 18) і виконується запит на додавання користувача в базу (№ 19–20).

```
9      <?
10      require("cnt.php");
11      if(!$als || !$psw || !$name || !$mname || !$lname ||
!$add)
12          tst_exit($msg_err_call);
13      $db = tst_connect_db($msg_err_cnt . $msg_not_reg);
14      $q = "SELECT usr_als FROM usr WHERE usr_als = '$als'";
15      $r = tst_exec_sql($q,$db,$msg_err_cnt . $msg_not_reg);
16      if (mysql_num_rows($r) > 0)
17          tst_exit("<h1                                align='center'><font
color='#FF0000'>Користувач      '$als'      вже      зареєстрований!!!
</font></h1>");
18      mysql_free_result($r);
19      $q = "INSERT INTO usr VALUES (NULL, '$als', '$psw',
'$name', '$mname', '$lname', '$add')";
20      $r = tst_exec_sql($q,$db,$msg_err_cnt . $msg_not_reg);
```

У № 21–60 виводиться інформація про нового (щойно зареєстрованого) користувача.

```
21      echo "
22      <table      width='100%'      border='0'      cellpadding='5'
cellpadding='0'>
23          <tr>
24              <td      align='center'><font      color='#0000FF'
size='+2'><b><i>Новий      користувач      зареєстрований
</i></b></font></td>
25          </tr>
26      </table>
27      <hr>
28      <table      width='400'      border='0'      cellpadding='5'
cellpadding='5' align='center'>
29          <tr>
30              <td width='102'><b>Псевдо</b></td>
31              <td width='278' valign='top' bgcolor='#CCCCCC'>
32                  <font color='#0000FF'><b>$als</b></font></td>
33          </tr>
```

```

34     <tr>
35         <td width='102'><b>Пароль</b></td>
36         <td width='278' valign='top' bgcolor='#CCCCCC'>
37             <font color='#0000FF'><b>$psw</b></font></td>
38     </tr>
39     <tr>
40         <td width='102'><b>Прізвище</b></td>
41         <td width='278' valign='top' bgcolor='#CCCCCC'>
42             <font color='#0000FF'><b>$lname</b></font></td>
43     </tr>
44     <tr>
45         <td width='102'><b>Ім'я</b></td>
46         <td width='278' valign='top' bgcolor='#CCCCCC'>
47             <font color='#0000FF'><b>$name</b></font></td>
48     </tr>
49     <tr>
50         <td width='102'><b>По-батькові</b></td>
51         <td width='278' valign='top' bgcolor='#CCCCCC'>
52             <font color='#0000FF'><b>$mname</b></font></td>
53     </tr>
54     <tr align='center'>
55         <td colspan='2'><b>Додаткова інформація</b></td>
56     </tr>
57     <tr align='center' valign='top' bgcolor='#CCCCCC'>
58         <td colspan='2'><font
color='#0000FF'><b>$add</b></font></td>
59     </tr>
60 </table>";
61 ?>

```

У № 62–67 формується посилання на головну сторінку (№ 65).

```

62     <hr>
63     <table width="100%" border="0" cellpadding="5"
cellpadding="0">
64         <tr>
65             <td align="center"><font size="+1"><a
href="index.html">Перехід на головну сторінку</a></font></td>
66         </tr>
67     </table>
68 </body>
69 </html>

```

Результат роботи програми (файлу) newuser.php:

```

1     <html>
2     <head>
3     <title>Новий користувач</title>
4     <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251; charset=windows-1251">

```

```

5      <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6      --></style>
7      </head>
8      <body bgcolor="#FFFFFF" text="#000000">
9
10     <table width='100%' border='0' cellspacing='5'
cellpadding='0'>
11         <tr>
12             <td align='center'><font color='#0000FF'
size='+2'><b><i>Новий користувач зареєстрований
</i></b></font></td>
13         </tr>
14     </table>
15     <hr>
16     <table width='400' border='0' cellspacing='5'
cellpadding='5' align='center'>
17         <tr>
18             <td width='102'><b>Псевдо</b></td>
19             <td width='278' valign='top' bgcolor='#CCCCCC'>
20                 <font color='#0000FF'><b>petrov</b></font></td>
21         </tr>
22         <tr>
23             <td width='102'><b>Пароль</b></td>
24             <td width='278' valign='top' bgcolor='#CCCCCC'>
25                 <font color='#0000FF'><b>petrov</b></font></td>
26         </tr>
27         <tr>
28             <td width='102'><b>Прізвище</b></td>
29             <td width='278' valign='top' bgcolor='#CCCCCC'>
30                 <font color='#0000FF'><b>Іващенко</b></font></td>
31         </tr>
32         <tr>
33             <td width='102'><b>Ім'я</b></td>
34             <td width='278' valign='top' bgcolor='#CCCCCC'>
35                 <font color='#0000FF'><b>Петро</b></font></td>
36         </tr>
37         <tr>
38             <td width='102'><b>По-батькові</b></td>
39             <td width='278' valign='top' bgcolor='#CCCCCC'>
40                 <font color='#0000FF'><b>Дмитрович</b></font></td>
41         </tr>
42         <tr align='center'>
43             <td colspan='2'><b>Додаткова інформація</b></td>
44         </tr>
45         <tr align='center' valign='top' bgcolor='#CCCCCC'>
46             <td colspan='2'><font color='#0000FF'><b>тестовий
користувач</b></font></td>
47         </tr>

```

```

48     </table>
49     <hr>
50     <table      width="100%"      border="0"      cellpadding="0">
51         <tr>
52             <td      align="center"><font      size="+1"><a
href="http://XXXXXXXXXX.ua/tests/index.html">Перехід      на      головну
сторінку</a></font></td>
53         </tr>
54     </table>
55 </body>
56 </html>

```

Для продовження роботи користувачеві необхідно перейти на головну сторінку.

Створення особистої сторінки користувача

Для відображення особистої сторінки користувача використовується файл (програма на мові php) `logon.php`.

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

```

1     <html>
2     <head>
3     <title>Особиста      сторінка      користувача      [Тестування
знань]</title>
4     <meta      http-equiv="Content-Type"      content="text/html;
charset=windows-1251">
5     <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6     --></style>
7     </head>
8     <body bgcolor="#FFFFFF" text="#000000">
9     <?

```

У № 10 вказується, що буде використаний файл `cnt.php` (див. вище).

У № 11 відбувається з'єднання з базою даних.

В № 12–30 здійснюється пошук у базі даних інформації про користувача.

Якщо інформація про номер (ідентифікатор) користувача передається в програму як параметр (змінна `$userid` має значення, що не дорівнює нулю), то в базі даних ведеться пошук інформації за номером користувача (№ 13–19). Текст запиту визначається в № 13–14. Виконання запиту здійснюється в № 15 за допомогою функції «`tst_exec_sql`».

У № 16 перевіряється коректність інформації про користувача в базі даних (запит повинен повертати одну запис), якщо дані є некоректними, то відбувається аварійний вихід і виведення повідомлення про помилку.

За допомогою вбудованої функції «php mysql_fetch_array» (№ 17) здійснюється перетворення записи з результату запиту в масив (значення в масиві можна знайти по імені змінної в SQL запиті).

```
10     require("cnt.php");
11     $db = tst_connect_db($msg_err_cnt);
12     if($userid){
13         $q="SELECT                                usr_id,usr_als,
usr_psw,usr_name,usr_mname,usr_lname
14         FROM usr WHERE usr_id = '$userid'";
15         $r = tst_exec_sql($q,$db,$msg_err_cnt);
16         if(mysql_num_rows($r) != 1 ) tst_exit($msg_err_call);
17         $usr = mysql_fetch_array($r);
18         $als = $usr["usr_als"];
19     }
```

Якщо інформація про номер (ідентифікатор) користувача не передається в програму як параметр (змінна \$userid дорівнює нулю або не визначена), то в базі даних введеться пошук інформації за псевдонімом користувача і паролем (№ 21–30). Текст запиту визначається в № 21–22. Виконання запиту здійснюється в № 23 за допомогою функції «tst_exec_sql».

У № 24 перевіряється коректність інформації про користувача в базі даних.

```
20     else{
21         $q                                ="SELECT                                usr_id,usr_als,
usr_psw,usr_name,usr_mname,usr_lname
22         FROM usr WHERE usr_als = '$als'";
23         $r = tst_exec_sql($q, $db, $msg_err_cnt);
24         if (mysql_num_rows($r) != 1 )
25             tst_exit("<h1                                align='center'><font
color='#FF0000'>Користувач                                '$als'                                не
zareestrovаний!!!</font></h1>");
26         $usr = mysql_fetch_array($r);
27         if($usr["usr_psw"] != $psw)
28             tst_exit($msg_err_psw);
29         $userid = $usr["usr_id"];
30     }
31     ?>
```

У № 32 виводиться заголовок сторінки. У № 34–37 виводиться ім'я користувача і його псевдо.

У № 38–42 з бази вибираються записи про всі тести користувача. SQL запит будується в № 38–41.

З таблиці tst вибираються поля tst_id (номер тесту), tst_a (відповіді тесту), tst_ok (правильні відповіді), tst_bal (кількість балів за правильну відповідь), tst_tm (час, витрачений на проходження тесту). З таблиці def вибирається поле def_name (назва тесту).

Набір записів повинен задовольняти одночасно таким умовам (№ 40):

- номер користувача в таблиці `tst` (`tst.usr_id`) повинен збігатися з номером користувача, особиста сторінка якого обробляється в цей момент (інформація збережена в змінній `$userid`);
- номер опису тесту в таблиці `tst` (`def_id`) повинен збігатися з таким же номером в таблиці `def`.

Записи сортуються в порядку зростання номера тесту (№ 41).

Результат виконання запиту зберігається в змінній `$r` (№ 42), а кількість повернутих записів у змінній `$n_tot`.

У № 43–53 визначається кількість тестів (спроб) для користувача.

У змінну `$n_tot` записується число спроб з бази даних (№ 43), а в змінну `$n_res` – число «результативних» (тест зданий) спроб (№ 44).

Якщо є новий тест (це визначається в № 48–50), то в змінну `$new_tstid` записується номер нового тесту, а в змінну `$new_defname` – його назва.

```
32      <h1 align='center'><font color='#0000FF'>Особиста
сторінка.</font></h1>
33      <?
34      $nm = $usr["usr_name"];
35      $nmn = $usr["usr_mname"];
36      $lnm = $usr["usr_lname"];
37      echo "<h2 align='center'> $lnm $nm $nmn (<font
color='#0000FF'>$als</font>).</h2><hr>";
38      $q = "SELECT tst_id,tst_a,tst_ok,tst_bal,tst_tm,def_name
39          FROM tst,def
40          WHERE  tst.usr_id = $userid and  tst.def_id =
def.def_id
41          ORDER BY tst.tst_id";
42      $r = tst_exec_sql($q,$db,$msg_err_cnt);
43      $n_tot = mysql_num_rows($r);
44      $n_res = 0;
45      $new_tstid = 0;
46      for($i=0; $i<$n_tot; $i++){
47          $row = mysql_fetch_array($r);
48          if($row["tst_a"] == NULL){
49              $new_tstid=$row["tst_id"];
50              $new_defname = $row["def_name"];
51          }
52          else $n_res++;
53      }
```

Якщо новий тест не виявлено, то виводиться форма, яка пропонує підготувати новий тест (№ 54–73). Форма називається `frm_def` (№ 59), дані форми передаються програмі-обробнику (файлу) `newtest.php`.

Дані про занесені в базу типи тестів вибираються з таблиці `def` (№ 55). Якщо в базі немає жодного опису, то виводиться повідомлення про помилку

(№ 58). Користувачеві пропонується вибрати тип тексту зі списку допустимих значень, який формується в циклі (№ 63–68).

```
54     if($new_tstid == 0){
55         $def_res = tst_exec_sql("SELECT def_id,def_name FROM
def",$db,$msg_err_con);
56         $def_tot = mysql_num_rows($def_res);
57         if($def_tot < 1)
58             tst_exit("<h1 align='center'><font
color='#FF0000'>Немає описів тестів</font></h1>");
59         echo "<form name='frm_def' method='post'
action='newtest.php'>
60             <div align='center'>
61                 <font color='#0000FF' size='+1'><b>Новий
тест</b></font>
62                 <select name='defid'>";
63         for($i=0; $i<$def_tot; $i++){
64             $def_row = mysql_fetch_array($def_res);
65             $did = $def_row["def_id"];
66             $dnm = $def_row["def_name"];
67             echo "<option value='$did'>$dnm</option>";
68         }
69         echo "</select>
70             <input type='submit' value='Підготувати тест'>
71             <input type='hidden' name='userid'
value='$userid'>
72             </div></form><hr>";
73     }
```

Якщо новий тест виявлений, то виводиться форма, яка пропонує виконати тест (№ 76–80). Форма називається `form_back` (№ 76), дані форми передаються програмі-обробнику (файлу) `test.php`.

```
74     else{
75         echo "<h2 align='center'>Для Вас підготовлен тест<font
color='#0000FF'> ($new_defname).</font></h2>
76         <form name='form_back' method='post'
action='test.php'>
77             <div align='center'>
78                 <input type='submit' value='Виконати тест'>
79                 <input type='hidden' name='testid'
value='$new_tstid'>
80             </div></form><hr>";
81     }
```

Якщо користувач не здав жодного тесту, то відбувається виведення відповідного повідомлення і вихід з програми (№ 82–83). У цьому випадку вихід не є аварійним (помилково), незважаючи на те, що використовується функція попередження виходу.

Якщо користувач вже здав тести, то відбувається висновок інформації про всі його спроби (№ 84–138).

У № 84–91 виводиться заголовок таблиці спроб. Інформація про спроби виводиться в циклі (№ 93–125).

Вбудована функція «php mysql_data_seek» (№ 92) встановлює поточну позицію в наборі даних (записів), повернутому SQL запитом (див. нижче, № 42). параметр «0» позначає, що поточної стає перший запис.

У масив \$ arr_а заносяться номери відповідей на кожне питання (в базі вони записані в один рядок і розділені пробілами). Для розбиття рядка використовується вбудована функція «php split» (№ 96).

Аналогічно в масиви \$ arr_ok і \$ arr_bl заносяться правильні відповіді на питання (№ 97) і кількості балів, що нараховуються за правильну відповідь (№ 98).

Загальна кількість відповідей (і питань) запам'ятовується в змінній \$ an_tot (№ 99). Використовується вбудована функція «php count», яка визначає число елементів в масиві.

У циклі (№ 100–109) визначаються результати для кожної спроби. У змінну bl_ok записується кількість набраних балів, а в змінну \$ bl_tot – загальна кількість балів.

Оцінка за п'ятибальною шкалою записується в змінну \$ an_res (№ 110).

У № 111–114 визначається колір фону комірки таблиці, в яку виводиться оцінка. Визначено такі кольори:

- # 0099FF (синій) – відмінно;
- # 00CC99 (зелений) – добре;
- # FFFF33 (жовтий) – задовільно;
- # FF0000 (червоний) – незадовільно.

Дата записується в змінну \$ stm (№ 115), вона форматується за допомогою вбудованої функції «php date».

Виведення рядка таблиці здійснюється в № 116–124.

У кінці сторінки виводиться легенда, що описує колір фону для кожної оцінки (№ 128–138).

```
82     if($n_res < 1)
83         tst_exit("<h2 align='center'><font color='#0000FF'>Ви
ще не склали жодного теста.</font></h2>");
84     echo"<table width='100%' border='0' cellpadding='5'>
85         <tr bgcolor='#0000FF'>
86             <th bgcolor='#0000FF'><font color='#FFFFFF'
size='+1'>Тест</font></th>
87             <th><font color='#FFFFFF'
size='+1'>Дата</font></th>
88             <th><font color='#FFFFFF'
size='+1'>Кількість<br>питань</font></th>
```

```

89         <th><font                                color='#FFFFFF'
size='+1'>Максимальна<br>сума<br>балів</font></th>
90         <th><font                                color='#FFFFFF'
size='+1'>Сума<br>балів</font></th>
91         <th><font                                color='#FFFFFF'
size='+1'>Результат</font></th></tr>";
92     mysql_data_seek($r,0);
93     for($i=0; $i<$n_tot; $i++){
94         $row = mysql_fetch_array($r);
95         if($row["tst_a"] == NULL) continue;
96         $arr_a = split(" ",$row["tst_a"]);
97         $arr_ok = split(" ",$row["tst_ok"]);
98         $arr_bl = split(" ",$row["tst_bal"]);
99         $an_tot = count($arr_ok);
100        $an_ok = 0;
101        $bl_tot = 0;
102        $bl_ok = 0;
103        for($j=0; $j<$an_tot; $j++){
104            $bl_tot += $arr_bl[$j];
105            if($arr_a[$j] == $arr_ok[$j]){
106                $an_ok++;
107                $bl_ok += $arr_bl[$j];
108            }
109        }
110        $an_res = (double)$bl_ok/(double)$bl_tot*5.0;
111        if($an_res > 4.0) $color = "#0099FF";
112        elseif($an_res > 3.0) $color = "#00CC99";
113        elseif($an_res > 2.0) $color = "#FFFF33";
114        else $color = "#FF0000";
115        $stm = date("d.m.Y H:i:s",$row["tst_tm"]);
116        printf("
117            <tr align='center' valign='top'>
118            <td bgcolor='#CCCCCC'><b>%s</b></td>
119            <td bgcolor='#CCCCCC'><b>%s</b></td>
120            <td bgcolor='#CCCCCC'><b>%d</b></td>
121            <td bgcolor='#CCCCCC'><b>%d</b></td>
122            <td bgcolor='#CCCCCC'><b>%d</b></td>
123            <td bgcolor='%s'><b>%5.2f</b></td>",
124            $row["def_name"],$stm,$an_tot,$bl_tot,
125            $bl_ok,$color,$an_res);
126        }
127        echo "</table><hr>";
128        ?>
129        <table width="100%" border="0" cellpadding="10">
130            <tr>
131                <td colspan="4" align="center"><font
size="+1"><i>Легенда</i></font></td>
132            </tr>
133            <tr align="center">

```

```

133         <td          bgcolor="#0099FF"          align="center"
valign="top"><b>Відмінно</b></td>
134         <td          bgcolor="#00CC99"          align="center"
valign="top"><b>Добре</b></td>
135         <td  bgcolor="#FFFF33"  align="center"  valign="top"
width="25%"><b>Задовільно</b></td>
136         <td  bgcolor="#FF0000"  align="center"  valign="top"
width="25%"><b>Незадовільно</b></td>
137     </tr>
138 </table>
139 </body>
140 </html>

```

У файлі `logon_php1.html` наведено результат роботи програми `logon.php` в разі, коли користувач не здав жодного тесту і йому видається форма підготовки нового тесту.

```

1  <html>
2  <head>
3  <title>Особиста сторінка користувача [Тестування
знань]</title>
4  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251; charset=windows-1251">
5  <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6  --></style>
7  </head>
8  <body bgcolor="#FFFFFF" text="#000000">
9  <h1 align='center'><font color='#0000FF'>Особиста
сторінка.</font></h1>
10 <h2 align='center'> Козловський Іван Семенович (<font
color='#0000FF'>kozlovskiy</font>).</h2>
11 <hr>
12 <form name='frm_def' method='post'
action='http://XXXXXXXXXX.ua/tests/newtest.php'>
13     <div align='center'>
14         <font color='#0000FF' size='+1'><b>Новий
тест</b></font>
15         <select name='defid'>
16             <option value='1'>Microsoft Word</option>
17             <option value='2'>Microsoft Excel</option>
18         </select>
19         <input type='submit' value='Підготувати тест'>
20         <input type='hidden' name='userid' value='104'>
21     </div>
22 </form>
23 <hr>
24 <h2 align='center'><font color='#0000FF'>Ви ще не склали
жодного теста.</font></h2>
25 </body>
26 </html>

```

У файлі `logon_php2.html` наведено результат роботи програми `logon.php` в разі, коли користувач не здав жодного тесту і для нього вже підготовлений новий тест.

```
1    <html>
2    <head>
3    <title>    Особиста    сторінка    користувача    [Тестування
знань]</title>
4    <meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1251; charset=windows-1251">
5    <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
7    </head>
8    <body bgcolor="#FFFFFF" text="#000000">
9    <h1 align='center'><font    color='#0000FF'>Особиста
сторінка.</font></h1>
10   <h2 align='center'>    Козловський    Іван    Семенович    (<font
color='#0000FF'> kozlovskyi </font>).</h2>
11   <hr>
12   <h2 align='center'>Для Вас підготовлений тест
13   <font color='#0000FF'> (Microsoft Word).</font></h2>
14   <form                name='form_back'                method='post'
action='http://XXXXXXXXXX.ua/tests/test.php'>
15       <div align='center'>
16           <input type='submit' value='Виконати тест'>
17           <input type='hidden' name='testid' value='385'>
18       </div>
19   </form>
20   <hr>
21   <h2 align='center'><font color='#0000FF'>Ви ще не склали
жодного тесту.</font></h2>
22   </body>
23   </html>
```

У файлі `logon_php3.html` наведено результат роботи програми `logon.php` в разі, коли користувач вже здав хоча б один тест і йому видається форма підготовки нового тесту.

```
1    <html>
2    <head>
3    <title>    Особиста    сторінка    користувача    [Тестування
знань]</title>
4    <meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1251; charset=windows-1251">
5    <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
7    </head>
8    <body bgcolor="#FFFFFF" text="#000000">
9    <h1 align='center'><font    color='#0000FF'>Особиста
сторінка.</font></h1>
```

```

10    <h2 align='center'> Козловський Іван Семенович (<font
color='#0000FF'>kozlovskiy</font>).
11    </h2>
12    <hr>
13    <form                name='frm_def'                method='post'
action='http://XXXXXXXXX.ua/tests/newtest.php'>
14        <div align='center'>
15            <font                color='#0000FF'                size='+1'><b>Новий
тест</b></font>
16                <select name='defid'>
17                    <option value='1'>Microsoft Word</option>
18                    <option value='2'>Microsoft Excel</option>
19                </select>
20                <input type='submit' value='Підготувати тест'>
21                <input type='hidden' name='userid' value='104'>
22            </div>
23    </form>
24    <hr>
25    <table width='100%' border='0' cellpadding='5'>
26        <tr bgcolor='#0000FF'>
27            <th          bgcolor='#0000FF'><font                color='#FFFFFF'
size='+1'>Тест</font></th>
28            <th><font color='#FFFFFF' size='+1'>Дата</font></th>
29            <th><font                color='#FFFFFF'
size='+1'>Кількість<br>питань</font></th>
30            <th><font                color='#FFFFFF'
size='+1'>Максимальна<br>сума<br>балів</font></th>
31            <th><font                color='#FFFFFF'
size='+1'>Сума<br>балів</font></th>
32            <th><font                color='#FFFFFF'
size='+1'>Результат</font></th>
33        </tr>
34        <tr align='center' valign='top'>
35            <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
36            <td          bgcolor='#CCCCCC'><b>26.12.2010
08:19:46</b></td>
37            <td bgcolor='#CCCCCC'><b>20</b></td>
38            <td bgcolor='#CCCCCC'><b>56</b></td>
39            <td bgcolor='#CCCCCC'><b>3</b></td>
40            <td bgcolor='#FF0000'><b> 0.27</b></td>
41        </tr>
42    </table>
43    <hr>
44    <table width="100%" border="0" cellpadding="10">
45        <tr>
46            <td                colspan="4"                align="center"><font
size="+1"><i>Легенда</i></font></td>
47        </tr>
48        <tr align="center">

```

```

49         <td          bgcolor="#0099FF"          align="center"
valign="top"><b>Відмінно</b></td>
50         <td          bgcolor="#00CC99"          align="center"
valign="top"><b>Добре</b></td>
51         <td  bgcolor="#FFFF33"  align="center"  valign="top"
width="25%"><b>Задовільно</b></td>
52         <td  bgcolor="#FF0000"  align="center"  valign="top"
width="25%"><b>Незадовільно</b></td>
53     </tr>
54 </table>
55 </body>
56 </html>

```

Створення нового тесту

Для створення нового тесту використовується файл (програма на мові php) newtest.php.

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

```

1     <html>
2     <head>
3     <title>Новий тест</title>
4     <meta  http-equiv="Content-Type"  content="text/html;
charset=windows-1251">
5     <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6     --></style>
7     </head>
8     <body bgcolor="#FFFFFF"  text="#000000"  leftmargin="0"
topmargin="0"  marginwidth="0"  marginheight="0">

```

У № 11 ініціалізується генератор випадкових чисел за допомогою вбудованої функції «php srand». Як ініціюючого числа використовується кількість секунд, що пройшли з початку Епохи Unix (1 січня 1970). Для отримання поточного часу використовується вбудована функція «php microtime».

У № 13 вказується, що буде використаний файл cnt.php (див. вище).

У № 15–16 перевіряється правильність переданих параметрів (необхідні ім'я користувача та ідентифікатор опису тесту).

У № 18 відбувається з'єднання з базою даних.

У № 18–21 з таблиці опису тестів (def) вибирається інформація відповідно до згенерованого тесту (ідентифікатор записаний в змінній \$ defid).

За допомогою вбудованої функції «php mysql_fetch_array» (№ 22) здійснюється перетворення записи з результату запиту в масив \$ def.

У № 23 правила побудови тесту (поле def_rule таблиці def) розшифровуються і записуються в масив \$ rule.

Нагадаємо, що в поле `def_rule` таблиці `def` зберігається така інформація: номер групи питань, пробіл, кількість питань із групи в тесті. Інформація вводиться для кожної групи (роздільник – символ `#`). Для розбиття рядка використовується вбудована функція `«php split»`.

У № 24 за допомогою вбудованої функції `«php mysql_free_result»` звільняється пам'ять від результату запиту, виконаного в № 21.

У циклі (№ 25–29) у масив `$ gr_gr` записується номер групи, а в масив `$ gr_n` – кількість питань у групі. Допоміжний масив `$ vvv` видаляється з пам'яті за допомогою вбудованої функції `«php unset»` (№ 30).

```
10  <?
11  srand((double)microtime()*1000000);
13  require("cnt.php");
15  if(!$userid || !$defid)
16      tst_exit($msg_err_call);
18  $db = tst_connect_db($msg_err_cnt);
20  $q = "SELECT * FROM def WHERE def_id = $defid";
21  $r = tst_exec_sql($q, $db, $msg_err_cnt);
22  $def = mysql_fetch_array($r);
23  $rule = split("#", $def["def_rule"]);
24  mysql_free_result($r);
25  for($i=0; $i < count($rule); $i++){
26      $vvv = split(" ", $rule[$i]);
27      $gr_gr[$i] = $vvv[0];
28      $gr_n[$i] = $vvv[1];
29  }
30  unset($vvv);
```

У циклі (№ 33–61) генеруються питання тесту.

У № 34–36 з таблиці `q` вибираються всі питання групи, номер якої записаний в масиві `$ gr_gr`. Вибирається така інформація:

- з поля `q_id` – номер питання;
- з поля `q_ok` – номер правильної відповіді;
- з поля `q_bal` – кількість балів за правильну відповідь.

Кількість питань у групі зберігається в змінній `$ num_r` (№ 37).

У циклі (№ 38–41) інформація про питаннях кожної групи записується в масив `$ vvv`. Кожен елемент цього масиву – рядок, у якій записуються номер питання, номер правильної відповіді і кількість балів за правильну відповідь (як роздільник використовується пропуск).

За допомогою вбудованої функції `«php shuffle»` елементи масиву `$ vvv` перемішуються у випадковому порядку (№ 42).

У циклі (№ 43–46) з масиву `$ vvv` вибираються питання кожної групи і записуються в загальний для всього тесту масив `$ arr_qw_ok` (№ 44).

Допоміжний масив `$ vvv` видаляється з пам'яті за допомогою вбудованої функції `«php unset»` (№ 47), а результат запиту з таблиці `q` (виконаний в № 36) видаляється з пам'яті за допомогою вбудованої функції `«php mysql_free_result»` (№ 48).

За допомогою вбудованої функції «php shuffle» елементи масиву \$arr_qw_ok перемішуються у випадковому порядку (№ 51).

У № 53–61 відбувається підготовка інформації для запису в таблицю tst.

У № 63–64 інформація записується в базу даних. Поля таблиці tst заповнюються так:

- поле tst_id (номер тесту) має тип auto_increment і заповнюється автоматично. У тексті запиту в це поле записується NULL;
- у полі usr_id (номер користувача) записується значення змінної \$userid;
- у полі def_id (номер опису тесту) записується значення змінної \$defid;
- у полі tst_q (питання тесту) записується значення змінної \$str_qw;
- поле tst_a (відповіді тесту) залишається порожнім. У тексті запиту в це поле записується NULL;
- у полі tst_ok (правильні відповіді) записується значення змінної \$str_ok;
- у полі tst_bal (кількості балів за правильну відповідь) записується значення змінної \$str_bl;
- поле tst_tm (час, витрачений на проходження тесту) залишається порожнім. У тексті запиту в це поле записується число 0.

```
32     $cur_q = 0;
33     for($i=0; $i < count($gr_gr); $i++){
34         $q = "SELECT q_id,q_ok,q_bal FROM q
35             WHERE q_gr = $gr_gr[$i]";
36         $r = tst_exec_sql($q,$db,$msg_err_cnt);
37         $num_r = mysql_num_rows($r);
38         for($j=0; $j<$num_r; $j++){
39             $row = mysql_fetch_array($r);
40             $vvv[$j] = $row["q_id"]." ".$row["q_ok"]."
".$row["q_bal"];
41         }
42         shuffle($vvv);
43         for($j=0; $j<$gr_n[$i]; $j++){
44             $arr_qw_ok[$cur_q] = $vvv[$j];
45             $cur_q++;
46         }
47         unset($vvv);
48         mysql_free_result($r);
49     }
50
51     shuffle($arr_qw_ok);
52     $vvv = split(" ", $arr_qw_ok[0]);
53     $str_qw = $vvv[0];
54     $str_ok = $vvv[1];
55     $str_bl = $vvv[2];
56     for($i=1; $i<$cur_q; $i++){
57         $vvv = split(" ", $arr_qw_ok[$i]);
58         $str_qw = $str_qw ." ". $vvv[0];
59         $str_ok = $str_ok ." ". $vvv[1];
60         $str_bl = $str_bl ." ". $vvv[2];
61     }
```

```

63     $q = "INSERT INTO tst VALUES (NULL, $userid, $defid,
'$str_qw', NULL, '$str_ok', '$str_bl', 0)";
64     $r = tst_exec_sql($q,$db,$msg_err_cnt);
65

```

У № 66–73 виводиться форма повернення на особисту сторінку. Им'я форми «**form_back**», програма обробник – **logon.php**.

У прихованому (hidden) полі **userid** в обробник передається ідентифікатор користувача (№ 71).

```

66     echo"
67     <h1 align='center'><font color='#0000FF'>Підготовлено
Новий тест</font></h1>
68     <form name='form_back' method='post' action='logon.php'>
69         <div align='center'>
70             <input type='submit' value='Повернутися на особисту
сторінку'>
71             <input type='hidden' name='userid' value='$userid'>
72         </div>
73     </form>";
75     ?>
76 </body>
77 </html>

```

У файлі **newtest_php.html** наведено результат роботи програми **newtest.php**.

```

1     <html>
2     <head>
3     <title>Новий тест</title>
4     <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251; charset=windows-1251">
5     <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6     --></style>
7     </head>
8     <body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0">
9     <h1 align='center'><font color='#0000FF'>Підготовлено
новий тест</font></h1>
10    <form name='form_back' method='post'
action='http://XXXXXXXXXX.ua/tests/logon.php'>
11        <div align='center'>
12            <input type='submit' value='Повернутися на особисту
сторінку'>
13            <input type='hidden' name='userid' value='104'>
14        </div>
15    </form>
16 </body>
17 </html>

```

Виконання тесту

Тест виконується за допомогою файла (програми мовою php) **test.php**.

№ 1–6 описують стандартний початок html-сторінки (№ 5–6 описують стилі, що використовуються).

```
1    <html>
2    <head>
3    <title>Тестування знань [питання]</title>
4    <meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1251">
5    <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
```

У № 9 вказується, що буде використаний файл `cnt.php` (див. вище).

У №11–12 перевіряється, чи був переданий програмі ідентифікатор (номер) тесту. Якщо ідентифікатора немає (змінна `$ testid` не визначена або дорівнює нулю), то відбувається аварійний вихід і видається повідомлення про помилку.

У № 15 відбувається з'єднання з базою даних.

Якщо параметр «`$ hgst_tot`» (всього питань у тексті) не визначений або дорівнює нулю, то формується перше питання (№ 17–26).

З таблиці `tst` вибирається рядок, у якій записані номери питань (через пробіл) тесту (№ 18–20). Рядок розбивається за допомогою вбудованої функції «`php split`», результат заноситься в масив `$ arr_q` (№ 21).

Загальна кількість питань у тесті (кількість елементів у масиві `$ arr_q`) записується в змінну `$ hgst_tot` (№ 22). Рядок номерів питань записується в змінну `$ hgst_q` (№ 23). Номер поточного питання стає рівним 1 і записується в змінну `$ hgst_n` (№ 24).

У № 25 видаляється з пам'яті результат запиту до таблиці `tst` (виконується в № 19).

```
8    <?
9    require("cnt.php");
11   if(!$testid)
12       tst_exit($msg_err_call);
13   if(!$ans)$ans=0;
15   $db = tst_connect_db($msg_err_cnt);
17   if(!$hgst_tot){
18       $q = "SELECT tst_q FROM tst WHERE tst_id = $testid";
19       $r = tst_exec_sql($q,$db,$msg_err_cnt);
20       $row = mysql_fetch_array($r);
21       $arr_q = split(" ", $row["tst_q"]);
22       $hgst_tot = count($arr_q);
23       $hgst_q = $row["tst_q"];
24       $hgst_n = 1;
25       mysql_free_result($r);
26   }
```

Якщо параметр «\$ hgst_n» (номер поточного питання) дорівнює \$ hgst_tot (всього питань у тексті), то виконання тесту завершується (№ 27–46).

В іншому випадку виконання тесту триває (№ 47–58). У кінець строкової змінної \$ hqst_a дописується відповідь на питання з змінної \$ ans (№ 48–49). Номер поточного питання (\$ hgst_n) збільшується на одиницю (№ 50).

У масив \$ arr_q (№ 53) записуються ідентифікатори питань тесту. У № 54 визначається ідентифікатор поточного питання, його значення записується в змінну \$ cur_qid.

З таблиці q вибирається вся інформація про питання з номером (ідентифікатором) \$ cur_qid (№ 56–57) і записується в масив \$ tst (№ 58).

Завершення тесту супроводжується такими діями.

У кінець строкової змінної \$ hqst_a дописується відповідь на питання з змінної \$ ans (№ 28–29).

У змінну \$ tm за допомогою вбудованої функції «php time» записується поточний час (№ 30).

У таблиці tst оновлюється запис, який має номер \$ testid (№ 31–32). У записі поле tst_a стає рівним \$ hqst_a, а поле tst_tm – \$ tm.

З таблиці tst зчитується ідентифікатор користувача, який виконує тест (33–35). Це значення записується в змінну \$ userid (№ 36).

У № 37–44 виводиться форма повернення на особисту сторінку. Форма називається form_back, програма-обробник logon.php (№ 39). Номер користувача, який виконав тест передається в прихованому полі форми userid (№ 42).

У № 43 здійснюється завершення роботи програми.

```
27     elseif($hqst_n == $hqst_tot){
28         if(!$hqst_a) $hqst_a = $hqst_a . $ans;
29         else $hqst_a = $hqst_a ." ". $ans;
30         $tm = time();
31         $q = "update tst set tst_a='$hqst_a', tst_tm=$tm where
tst_id = $testid";
32         $r = tst_exec_sql($q,$db,$msg_err_cnt);
33         $q = "select usr_id from tst where tst_id = $testid";
34         $r = tst_exec_sql($q,$db,$msg_err_cnt);
35         $tst = mysql_fetch_array($r);
36         $userid = $tst["usr_id"];
37     echo"
38     <h1 align='center'><font color='#0000FF'>Виконання тесту
завершено</font></h1>
39     <form name='form_back' method='post' action='logon.php'>
40         <div align='center'>
41             <input type='submit' value=Повернутися до особистої
сторінки'>
42             <input type='hidden' name='userid' value='$userid'>
43         </div>
44     </form>";
45     exit;
46 }
```

```

47     else{
48         if($hqst_a == "") $hqst_a = $hqst_a . $ans;
49         else $hqst_a = $hqst_a ." ". $ans;
50         $hqst_n++;
51     }
53     $arr_q = split(" ", $hqst_q);
54     $cur_qid = $arr_q[$hqst_n-1];
56     $q = "select * from q where q_id = $cur_qid";
57     $r = tst_exec_sql($q, $db, $msg_err_cnt);
58     $tst = mysql_fetch_array($r);
59     ?>
60

```

У № 61–78 описана функція, яка контролює час відповіді на кожне питання. Функція написана на мові програмування JavaScript і називається «qstTimerFunc» (№ 67).

Час, який відводиться на питання збережено в масиві \$ tst. Це значення присвоюється змінної JavaScript qstTime (№ 63–65).

Функція «qstTimerFunc» прив'язується до системного таймеру за допомогою методу (функції) setTimeout об'єкта JavaScript window (№ 75). Таймер спрацьовує кожну секунду (1 000 мілісекунд). Інформація про час, що залишився виводиться в рядок стану браузера (№ 74). Потім залишився зменшується на секунду (№ 76).

Якщо час минув (№ 68–73), то таймер відключається (№ 69), очищається рядок стану браузера (№ 70) і ініціюється подія передачі даних серверу форми frmAns (див. вище № 103–137). Потім відбувається вихід їхньої функції (№ 72).

Функція «qstTimerFunc» викликається при завантаженні сторінки, вона прив'язана до події onLoad (№ 82).

```

61     <script language="javascript">
62     <?
63     $t=$tst["q_tm"];
64     echo "var qstTime = $t;
65     ";
66     ?>
67     function qstTimerFunc() {
68         if(qstTime == 0){
69             window.clearTimeout(qstTO);
70             window.status = "";
71             document.frmAns.submit();
72             return;
73         }
74         window.status = "Залишилось "+qstTime+ " хв.(сек.)";
75         qstTO=window.setTimeout("qstTimerFunc();",1000);
76         qstTime--;
77     }
78     </script>
79     </head>

```

```

81 <body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0"
82 onLoad="qstTimerFunc()">

```

У № 83–102 виводиться номер питання в тесті, загальна кількість питань і час, відведений на відповідь.

```

83 <table width="100%" border="0" cellspacing="5"
cellpadding="5">
84 <tr>
85 <td align="left">
86 <font color="#000000">
87 <b>Питання
88 <? echo $hqst_n; ?>
89 (всього
90 <? echo $hqst_tot; ?>
91 )</b>
92 </font></td>
93 <td align="right">
94 <font color="#000000">
95 <b>Час -
96 <? echo $tst["q_tm"]; ?>
97 хв.(сек)</b>.
98 </font></td>
99 </tr>
100
101 </table>
102 <hr>

```

Далі (№ 103–137) описана форма, яка виводить текст питання і можливі відповіді. Користувач повинен вибрати відповідь і натиснути кнопку «підтвердити відповідь». Після цього дані відправляються на сервер.

Ім'я форми – frmAns (№ 103). Програма обробник – файл test.php.

У № 104–111 виводиться текст питання, в № 112–124 – можливі відповіді. Можливий вибір тільки однієї відповіді (використовується елемент Radio Button).

Всі можливі відповіді записуються в масив \$ arr, для розбиття рядка використовується вбудована функція «php split» (№ 113). Потім вони виводяться в циклі (№ 114–123).

Після формування списку можливих відповідей виводиться кнопка підтвердження відповіді (№ 130) і заповнюються приховані поля, які використовуються для передачі інформації обробнику (№ 131–135). Передається така інформація:

- testid – номер користувача;
- hqst_tot – загальна кількість питань;
- hqst_n – номер поточного питання;
- hqst_q – питання тесту;
- hqst_a – відповіді.

```

103     <form name="frmAns" method="post" action="test.php">
104         <table width="100%" border="0" cellspacing="5"
cellpadding="5">
105             <tr>
106                 <td bgcolor="#CCCCCC" width="100" align="center"
valign="middle"><font size="+2"><b><font face="Times New Roman,
Times, serif" size="+4">?</font></b></font></td>
107                 <td width="*" align="left" valign="middle"><i><b>
108     <? echo $tst["q_q"]; ?>
109                 </b></i>
110             </td></tr>
111     <tr><td colspan='2' align='right' valign='middle'
bgcolor='#CCCCCC'>&nbsp;</td></tr>
112     <?
113     $arr = split("##",$tst["q_a"]);
114     while(list($k,$v) = each($arr)){
115         $kk=$k+1;
116         echo"<tr>
117             <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
118                 Відповідь $kk
119                 <input type='radio' name='ans' value='$kk'></b>
120             </td>
121             <td width='*'><b><i>$v</i></b></td>
122             </tr>";
123     }
124     ?>
127 </table>
128 <table width="100%" border="0" cellspacing="5"
cellpadding="0">
129     <tr><td align="center">
130     <input type="submit" value="Підтвердити відповідь">
131     <input type="hidden" name="testid" value=" <? echo
$ttestid ?>">
132     <input type="hidden" name="hqst_tot" value="<? echo
$hqst_tot ?>">
133     <input type="hidden" name="hqst_n" value="<? echo $hqst_n
?>">
134     <input type="hidden" name="hqst_q" value="<? echo $hqst_q
?>">
135     <input type="hidden" name="hqst_a" value="<? echo $hqst_a
?>">
136     </td></tr></table>
137 </form>
138 </body>
139 </html>

```

У файлі test_php1.html наведено перше питання тесту (користувач починає виконувати тест).

```

1    <html>
2    <head>
3    <title>Тестування знань [питання]</title>
4    <meta    http-equiv="Content-Type"    content="text/html;
charset=windows-1251; charset=windows-1251">
5    <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{    font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
7    <script language="javascript">
8    var qstTime = 30;
9    function qstTimerFunc() {
10    if(qstTime == 0){
11    window.clearTimeout(qstTO);
12    window.status = "";
13    document.frmAns.submit();
14    return;
15    }
16    window.status = "Залишилось "+qstTime+ " хв.(сек.)";
17    qstTO=window.setTimeout("qstTimerFunc();",1000);
18    qstTime--;
19    }
20    </script>
21    </head>
22    <body    bgcolor="#FFFFFF"    text="#000000"    leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0"
23    onLoad="qstTimerFunc()">
24    <table    width="100%"    border="0"    cellpadding="5"
cellspacing="5">
25    <tr>
26    <td align="left">
27    <font color="#000000">
28    <b>Питання
29    1(всього
30    20)</b>
31    </font></td>
32    <td align="right">
33    <font color="#000000">
34    <b>Час -
35    30 сек</b>.
36    </font></td>
37    </tr>
38    </table>
39    <hr>
40    <form    name="frmAns"    method="post"
action="http://XXXXXXXXXX.ua/tests/test.php">
41    <table    width="100%"    border="0"    cellpadding="5"
cellspacing="5">
42    <tr>

```

```

43         <td bgcolor="#CCCCCC" width="100" align="center"
valign="middle"><font size="+2"><b><font face="Times New Roman,
Times, serif" size="+4">?</font></b></font></td>
44         <td width="*" align="left" valign="middle"><i><b>
45     За допомогою якого елементу керування можна виділити
кольором тло?
46         </b></i>
47     </td></tr>
48     <tr><td colspan='2' align='right' valign='middle'
bgcolor='#CCCCCC'>&nbsp;  </td></tr>
49     <tr>
50         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
51             Відповідь 1
52             <input type='radio' name='ans' value='1'></b>
53         </td>
54         <td width='*'><b><i><img
src='tests_php1_files/wp110300.jpg'></i></b></td>
55     </tr><tr>
56         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
57             Відповідь 2
58             <input type='radio' name='ans' value='2'></b>
59         </td>
60         <td width='*'><b><i><img
src='tests_php1_files/wp110100.jpg'></i></b></td>
61     </tr><tr>
62         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
63             Відповідь 3
64             <input type='radio' name='ans' value='3'></b>
65         </td>
66         <td width='*'><b><i><img
src='tests_php1_files/wp110200.jpg'></i></b></td>
67     </tr><tr>
68         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
69             Відповідь 4
70             <input type='radio' name='ans' value='4'></b>
71         </td>
72         <td width='*'><b><i><img
src='tests_php1_files/wp100400.jpg'></i></b></td>
73     </tr><tr>
74         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
75             Відповідь 5
76             <input type='radio' name='ans' value='5'></b>
77         </td>

```

```

78         <td                                width='*'><b><i><img
src='tests_php1_files/wp100300.jpg'></i></b></td>
79     </tr></table>
80
81 </table>
82 <table      width="100%"      border="0"      cellspacing="5"
cellpadding="0">
83     <tr><td align="center">
84     <input type="submit" value="Підтвердити відповідь">
85     <input type="hidden" name="testid" value="385">
86     <input type="hidden" name="hqst_tot" value="20">
87     <input type="hidden" name="hqst_n" value="1">
88     <input type="hidden" name="hqst_q" value="77 63 31 17 26
255 212 194 49 188 228 153 55 276 175 130 39 111 245 271">
89     <input type="hidden" name="hqst_a" value="">
90     </td></tr></table>
91 </form>
92 </body>
93 </html>

```

У файлі **test_php2.html** наведено четверте питання теста (користувач вже відповів на три питання, див. № 86, № 88).

```

1     <html>
2     <head>
3     <title>Тестування знань [питання]</title>
4     <meta      http-equiv="Content-Type"      content="text/html;
charset=windows-1251; charset=windows-1251">
5     <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6     --></style>
7     <script language="javascript">
8     var qstTime = 30;
9     function qstTimerFunc() {
10         if(qstTime == 0){
11             window.clearTimeout(qstTO);
12             window.status = "";
13             document.frmAns.submit();
14             return;
15         }
16         window.status = "Залишилось "+qstTime+ " хв.(сек.)";
17         qstTO=window.setTimeout("qstTimerFunc();",1000);
18         qstTime--;
19     }
20 </script>
21 </head>
22 <body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0"
23     onLoad="qstTimerFunc()">

```

```

24     <table      width="100%"      border="0"      cellpadding="5">
cellpadding="5">
25         <tr>
26             <td align="left">
27                 <font color="#000000">
28                 <b>Питання
29                 4 (всього
30                 20)</b>
31             </font></td>
32             <td align="right">
33                 <font color="#000000">
34                 <b>Час -
35                 30 сек</b>.
36             </font></td>
37         </tr>
38     </table>
39     <hr>
40     <form          name="frmAns"          method="post"
action="http://XXXXXXXXX.ua/tests/test.php">
41         <table      width="100%"      border="0"      cellpadding="5">
cellpadding="5">
42             <tr>
43                 <td bgcolor="#CCCCCC" width="100" align="center"
valign="middle"><font size="+2"><b><font face="Times New Roman,
Times, serif" size="+4">?</font></b></font></td>
44                 <td width="*" align="left" valign="middle"><i><b>
45                 <img src='tests_php2_files/wp050500.jpg' align=left> Що
можна зробити за допомогою цього елемента керування?
46                 </b></i>
47             </td></tr>
48             <tr><td colspan='2' align='right' valign='middle'
bgcolor='#CCCCCC'>&nbsp;</td></tr>
49             <tr>
50                 <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
51                 Відповідь 1
52                 <input type='radio' name='ans' value='1'></b>
53             </td>
54             <td width='*'><b><i>додати          таблицю          в
текст</i></b></td>
55         </tr><tr>
56             <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
57             Відповідь 2
58             <input type='radio' name='ans' value='2'></b>
59         </td>
60         <td width='*'><b><i>ввімкнути          панель
'Малювання'</i></b></td>
61     </tr><tr>

```

```

62         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
63             Відповідь 3
64             <input type='radio' name='ans' value='3'></b>
65         </td>
66         <td width='*'><b><i> Показати схему
документа</i></b></td>
67     </tr><tr>
68         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
69             Відповідь 4
70             <input type='radio' name='ans' value='4'></b>
71         </td>
72         <td width='*'><b><i>Розбити виділений фрагмент
тексту на колонки</i></b></td>
73     </tr><tr>
74         <td width='100' align='right' valign='middle'
bgcolor='#CCCCCC'><b>
75             Відповідь 5
76             <input type='radio' name='ans' value='5'></b>
77         </td>
78         <td width='*'><b><i> Додати таблицю Excel
</i></b></td>
79     </tr></table>
80 </table>
81 <table width="100%" border="0" cellspacing="5"
cellpadding="0">
82     <tr><td align="center">
83         <input type="submit" value="Підтвердити відповідь">
84         <input type="hidden" name="testid" value="385">
85         <input type="hidden" name="hqst_tot" value="20">
86         <input type="hidden" name="hqst_n" value="4">
87         <input type="hidden" name="hqst_q" value="77 63 31 17 26
255 212 194 49 188 228 153 55 276 175 130 39 111 245 271">
88         <input type="hidden" name="hqst_a" value="3 4 4">
89     </td></tr></table>
90 </form>
91 </body>
92 </html>

```

У файлі test_php_fin.html наведено результат роботи системи після завершення тесту.

```

1 <html>
2 <head>
3 <title>Тестування знань [питання]</title>
4 <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251; charset=windows-1251">
5 <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}

```

```

6      --></style>
7      <h1 align='center'><font color='#0000FF'>Виконання тесту
завершено</font></h1>
8      <form                name='form_back'                method='post'
action='http://XXXXXXXXX.ua/tests/logon.php'>
9          <div align='center'>
10             <input type='submit' value='Повернутися до особистої
сторінки'>
11             <input type='hidden' name='userid' value='104'>
12         </div>
13     </form>
14 </body>
15 </html>

```

Вхід адміністратора в систему

Адміністратор входить в систему за допомогою виклику файлу admin.html.

```

1      <html>
2      <head>
3          <title>Тестування знань</title>
4          <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
5          <style type="text/css"><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6          --></style>
7      </head>
8      <body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0">
9          <form                name="logonadmin"                method="post"
action="admin.php">
10             <table width="400" border="0" cellspacing="0"
 cellpadding="5" align="center">
11                 <tr align="center">
12                     <td colspan="2">
13                         <b><i><font                                size="+2"
color="#0000FF">Адміністратор</font></i></b></td>
14                     </tr>
15                     <tr>
16                         <td width="30%" align="right"><b>Псевдо</b></td>
17                         <td width="50%">
18                             <input type="text" name="als" size="16"
maxlength="16">
19                         </td>
20                     </tr>
21                     <tr>
22                         <td width="30%" align="right"><b>Пароль</b></td>
23                         <td width="50%">
24                             <input type="password" name="psw" size="16"
maxlength="16">

```

```
25         </td>
26     </tr>
27     <tr align="center">
28         <td colspan="2">
29             <input type="submit" value="Зареєструватися">
30             &nbsp;
31         </td>
32     </tr>
33 </table>
34 </form>
35 </body>
36 </html>
```

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

Форма реєстрації користувача (№ 9–33) називається `logonadmin`, дані форми передаються програмі-обробнику (файлу) `admin.php` (№ 9). Ім'я користувача заноситься в поле форми `als` (№ 18), а пароль – у полі `psw` (№ 24).

Сторінка адміністратора

Для відображення особистої сторінки адміністратора використовується файл (програма мовою php) `admin.php`.

№ 1–8 описують стандартний початок html-сторінки (№ 5–6 описують використовувані стилі).

У № 11 вказується, що буде використаний файл `cnt.php` (див вище).

У № 13 відбувається з'єднання з базою даних.

```
1    <html>
2    <head>
3    <title>Сторінка          Адміністратора          [Тестування
знань]</title>
4    <meta    http-equiv='Content-Type'    content='text/html;
charset=windows-1251'>
5    <style type='text/css'><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6    --></style>
7    </head>
8    <body bgcolor='#FFFFFF' text='#000000'>
10   <?
11   require("cnt.php");
13   $db = tst_connect_db($msg_err_cnt);
```

У № 15–16 перевіряється, чи передані програмі псевдонім і пароль адміністратора, якщо параметри не визначені, то відбувається вихід помилково.

У № 17–19 з таблиці `usr` вибирається інформація про адміністратора системи (ідентифікатор користувача дорівнює 1), дані записується в масив `$adm` (№ 20).

У № 21–22 перевіряється правильність псевдоніма і пароля адміністратора, у разі помилки відбувається аварійний вихід.

У № 23 результати запиту будуть видалені з пам'яті.

```
15   if(!$als || !$psw)
16       tst_exit($msg_err_call);
17   $q = "SELECT usr_als, usr_psw
18       FROM usr WHERE usr_id = 1";
19   $r = tst_exec_sql($q, $db, $msg_err_cnt);
20   $adm = mysql_fetch_array($r);
21   if(($adm["usr_als"]!=$als) || ($adm["usr_psw"] !=$psw))
22       tst_exit($msg_err_psw);
23   mysql_free_result($r);
24   ?>
```

У № 26–30 виводиться заголовок сторінки. У № 28 встановлюється внутрішнє посилання на список користувачів.

У № 32 – 52 виводиться список користувачів.

Спочатку з таблиці `usr` вибирається інформація про всіх користувачів, крім адміністратора (номер користувача не дорівнює 1). Текст запиту формується в № 33–34, а виконується він в № 35. Результат запиту записується в змінну `$r_usr`.

Потім визначається кількість користувачів (№ 36), і в циклі (№ 37–49) виводиться список користувачів. Інформація про користувача записується у допоміжні змінні `$v1 – $v6` (№ 40–45). Виведення кожного рядка списку користувачів здійснюється в № 46–48. Для кожного користувача визначається внутрішнє посилання (№ 46) на детальну інформацію (див. нижче).

```
26      <h1 align='center'><font color='#0000FF'>Стопінка
Адміністратора</font></h1>
27      <hr>
28      <a name='usr'></a>
30      <table width='100%' border='0' cellpadding='5'
cellspacing='5'>
32      <?
33      $q="SELECT usr_id, usr_als, usr_psw, usr_name, usr_mname,
usr_lname, usr_add
34          FROM usr WHERE usr_id != 1";
35      $r_usr = tst_exec_sql($q,$db,$msg_err_cnt);
36      $usr_tot = mysql_num_rows($r_usr);
37      for($usr_n=0; $usr_n<$usr_tot; $usr_n++){
38          $usr = mysql_fetch_array($r_usr);
39          $userid = $usr["usr_id"];
40          $v1 = $usr["usr_name"];
41          $v2 = $usr["usr_mname"];
42          $v3 = $usr["usr_lname"];
43          $v4 = $usr["usr_als"];
44          $v5 = $usr["usr_add"];
45          $v6 = $usr["usr_psw"];
46          echo "<tr><td><b>$userid. <a href='#u$userid'>$v3 $v1
$v2</a>
47              (&quot;$v4&quot;;, &quot;$v6&quot;)</b></td>
48              <td bgcolor='#CCCCCC'><b>$v5</b></td></tr>";
49      }// of for($usr_n=0; $usr_n<$usr_tot; $usr_n++)
50      ?>
52      </table><hr>
```

Далі виводиться детальна інформація про користувача (результати всіх тестів, які він виконав).

У № 55 перший запис встановлюється поточним в наборі `$r_usr`.

У циклі (№ 55–145) виводиться інформація для кожного користувача. Інформація про користувача записується у допоміжні змінні (№ 58–64) і виводиться в № 66–76.

У № 66 встановлюється посилання для переходу зі списку користувачів.

```
54      <?
55      mysql_data_seek($r_usr,0);
```

```

56     for($usr_n=0; $usr_n<$usr_tot; $usr_n++){
57         $usr = mysql_fetch_array($r_usr);
58         $userid = $usr["usr_id"];
59         $v1 = $usr["usr_name"];
60         $v2 = $usr["usr_mname"];
61         $v3 = $usr["usr_lname"];
62         $v4 = $usr["usr_als"];
63         $v5 = $usr["usr_add"];
64         $v6 = $usr["usr_psw"];
66         echo"<a name='u$userid'></a>
67         <table width='100%' border='0' cellspacing='5'
cellpadding='5'><tr>
68             <td align='center'>
69                 <font color='#0000FF' face='Times New Roman, Times,
serif' size='+2'><b>$userid.</b></font>
70                 <b><font face='Times New Roman, Times, serif'
size='+2'> $v3 $v1 $v2
71                 (<font color='#0000FF'>&quot;$v4&quot;;
&quot;$v6&quot;</font>).</font></b>
72             </td></tr>
73             <tr>
74                 <td align='center'
bgcolor='#CCCCCC'><b>$v5</b></td>
75             </tr>
76         </table>";

```

У № 78–85 виводиться заголовок таблиці всіх тестів користувача.

```

78     echo"<table width='100%' border='0' cellspacing='5'>
79         <tr bgcolor='#0000FF'>
80             <th><font color='#FFFFFF' size='+1'>
Тест</font></th>
81             <th><font color='#FFFFFF' size='+1'>
Дата</font></th>
82             <th><font color='#FFFFFF' size='+1'>
Кол.<br>вопр.</font></th>
83             <th><font color='#FFFFFF' size='+1'>
Макс.<br>сумма<br>балов</font></th>
84             <th><font color='#FFFFFF' size='+1'>
Сумма<br>балов</font></th>
85             <th><font color='#FFFFFF' size='+1'>
Результат</font></th></tr>";

```

З таблиць `tst` і `def` вибирається інформація про всі тести користувача (№ 86–91). Дані виводяться в циклі (№ 92–135).

Якщо тест ще не виконано, то виводиться тільки назва тесту (№ 95–103). В іншому випадку виводиться повна інформація про тест.

Інформація витягується з кожного запису набору (№ 105–108), обчислюється загальна сума балів у цьому тесті і сума балів, яку набрав

користувач (№ 109–118). Потім визначається оцінка і колір фону комірки, який їй відповідає (№ 119–123).

У № 124 форматується дата і час проходження тесту.

Виведення рядка таблиці для кожного тесту здійснюється в № 125–133.

У № 137 виводиться посилання для повернення до списку користувачів.

```

86      $q = "SELECT tst_id, tst_a, tst_ok, tst_bal, tst_tm,
def_name
87          FROM tst,def
88          WHERE tst.usr_id = $userid and tst.def_id =
def.def_id
89          ORDER BY tst.tst_id";
90      $r = tst_exec_sql($q,$db,$msg_err_cnt);
91      $n_tot = mysql_num_rows($r);
92      for($i=0; $i<$n_tot; $i++){
93          $row = mysql_fetch_array($r);
94          if($row["tst_a"] == NULL)
95              printf("
96                  <tr align='center' valign='top'>
97                  <td bgcolor='#CCCCCC'><b>%s</b></td>
98                  <td>&nbsp;</td>
99                  <td>&nbsp;</td>
100                 <td>&nbsp;</td>
101                 <td>&nbsp;</td>
102                 <td>&nbsp;</td></tr>",
103                 $row["def_name"]);
104          else{
105              $arr_a = split(" ",$row["tst_a"]);
106              $arr_ok = split(" ",$row["tst_ok"]);
107              $arr_bl = split(" ",$row["tst_bal"]);
108              $an_tot = count($arr_ok);
109              $an_ok = 0;
110              $bl_tot = 0;
111              $bl_ok = 0;
112              for($j=0; $j<$an_tot; $j++){
113                  $bl_tot += $arr_bl[$j];
114                  if($arr_a[$j] == $arr_ok[$j]){
115                      $an_ok++;
116                      $bl_ok += $arr_bl[$j];
117                  }
118              }
119              $an_res = (double)$bl_ok/(double)$bl_tot*5.0;
120              if($an_res > 4.0) $color = "#0099FF";
121              elseif($an_res > 3.0) $color = "#00CC99";
122              elseif($an_res > 2.0) $color = "#FFFF33";
123              else $color = "#FF0000";
124              $stm = date("d.m.Y H:i:s",$row["tst_tm"]);
125              printf("
126                  <tr align='center' valign='top'>
```

```

127         <td bgcolor='#CCCCCC'><b>%s</b></td>
128         <td bgcolor='#CCCCCC'><b>%s</b></td>
129         <td bgcolor='#CCCCCC'><b>%d</b></td>
130         <td bgcolor='#CCCCCC'><b>%d</b></td>
131         <td bgcolor='#CCCCCC'><b>%d</b></td>
132         <td bgcolor='%s'><b>%5.2f</b></td>",
133         $row["def_name"], $stm, $an_tot, $bl_tot, $bl_ok,
$color, $an_res);
134     }
135 }
136 echo "</table><hr>";
137 echo "<div align='center'><a href='#usr'><b>Вернутися
к списку</b></a>";

```

Для кожного користувача виводиться легенда, що описує колір фону для кожної оцінки (№ 138–144).

```

138         <table width='100%' border='0' cellspacing='10'>
139             <tr align='center'>
140                 <td bgcolor='#0099FF' align='center'
valign='top'><b>Відмінно</b></td>
141                 <td bgcolor='#00CC99' align='center'
valign='top'><b>Добре</b></td>
142                 <td bgcolor='#FFFF33' align='center'
valign='top' width='25%'><b>Удовлетворительно</b></td>
143                 <td bgcolor='#FF0000' align='center'
valign='top' width='25%'><b>Неудовлетворительно</b></td>
144             </tr></table></div><hr>";
145     } // of for($usr_n=0; $usr_n<$usr_tot; $usr_n++)
146     ?>
147 </body>
148 </html>

```

У файлі `admin_php.html` наведено результат роботи програми `admin.php` для двох тестових користувачів.

```

1 <html>
2 <head>
3 <title>Сторінка Адміністратора [Тестування
знань]</title>
4 <meta http-equiv='Content-Type' content='text/html;
charset=windows-1251'>
5 <style type='text/css'><!-- A:HOVER {color: #FF0000} td
{ font-family: Arial, Helvetica, sans-serif; font-size: 14px}
6 --></style>
7 </head>
8 <body bgcolor='#FFFFFF' text='#000000'>
9 <h1 align='center'><font color='#0000FF'>Сторінка
Адміністратора</font></h1>
10 <hr>

```

```

11     <a name='usr'></a>
12     <table      width='100%'      border='0'      cellspacing='5'
cellpadding='5'>
13     <tr><td><b>2. <a href='#u2'>Петренко Петро Іванович</a>
14         (&quot;iv&quot;;, &quot;qwerty&quot;)</b></td>
15         <td
            bgcolor='#CCCCCC'><b>тестовий
користувач</b></td></tr>
16     <tr><td><b>3. <a href='#u3'> Петренко Петро Іванович </a>
17         (&quot;petrenko&quot;;,          &quot;petrenko&quot;)</b></td>
</b></td>
18         <td bgcolor='#CCCCCC'><b>студент</b></td></tr>
19     </table><hr>
20     <a name='u2'></a>
21     <table      width='100%'      border='0'      cellspacing='5'
cellpadding='5'><tr>
22         <td align='center'>
23         <font color='#0000FF' face='Times New Roman, Times,
serif' size='+2'><b>2.</b></font>
24         <b><font face='Times New Roman, Times, serif'
size='+2'> Вітер Валерій Петрович
25         (<font color='#0000FF'>&quot;iv&quot;;,
&quot;qwerty&quot;;</font>)</font></b>
26         </td></tr>
27         <tr>
28         <td align='center' bgcolor='#CCCCCC'><b>тестовий
користувач</b></td>
29         </tr>
30     </table><table      width='100%'      border='0'
cellspacing='5'>
31         <tr bgcolor='#0000FF'>
32         <th><font color='#FFFFFF'
size='+1'>Текст</font></th>
33         <th><font color='#FFFFFF'
size='+1'>Дата</font></th>
34         <th><font color='#FFFFFF'
size='+1'>Кількість<br>питань</font></th>
35         <th><font color='#FFFFFF'
size='+1'>Максимальна<br>сума<br>балів</font></th>
36         <th><font color='#FFFFFF'
size='+1'>Сума<br>балів</font></th>
37         <th><font color='#FFFFFF'
size='+1'>Результат</font></th></tr></table><hr><div
align='center'><a href='#usr'><b>Повернутися до списку</b></a>
38     <table width='100%' border='0' cellspacing='10'>
39         <tr align='center'>
40         <td bgcolor='#0099FF' align='center'
valign='top'><b>Відмінно</b></td>
41         <td bgcolor='#00CC99' align='center'
valign='top'><b>Добре</b></td>

```

```

42          <td          bgcolor='#FFFF33'          align='center'
valign='top' width='25%'><b>Задовільно</b></td>
43          <td          bgcolor='#FF0000'          align='center'
valign='top' width='25%'><b>Незадовільно</b></td>
44      </tr></table></div><hr>
45      <a name='u3'></a>
46      <table width='100%' border='0' cellspacing='5'
cellpadding='5'><tr>
47          <td align='center'>
48              <font color='#0000FF' face='Times New Roman, Times,
serif' size='+2'><b>3.</b></font>
49              <b><font face='Times New Roman, Times, serif'
size='+2'>Петрнеко Петро Іванович
50                  (<font color='#0000FF'>&quot;petrenko&quot;;,
&quot;petrenko&quot;</font>).</font></b>
51          </td></tr>
52      <tr>
53          <td align='center'
bgcolor='#CCCCCC'><b>студент</b></td>
54      </tr>
55      </table><table width='100%' border='0'
cellspacing='5'>
56          <tr bgcolor='#0000FF'>
57              <th><font color='#FFFFFF'
size='+1'>Тест</font></th>
58              <th><font color='#FFFFFF'
size='+1'>Дата</font></th>
59              <th><font color='#FFFFFF'
size='+1'>Кількість<br>питань</font></th>
60              <th><font color='#FFFFFF'
size='+1'>Максимальна<br>сума<br>балів</font></th>
61              <th><font color='#FFFFFF'
size='+1'>Сума<br>балів</font></th>
62              <th><font color='#FFFFFF'
size='+1'>Результат</font></th></tr>
63          <tr align='center' valign='top'>
64              <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
65              <td          bgcolor='#CCCCCC'><b>19.10.2010
19:31:34</b></td>
66              <td bgcolor='#CCCCCC'><b>20</b></td>
67              <td bgcolor='#CCCCCC'><b>56</b></td>
68              <td bgcolor='#CCCCCC'><b>47</b></td>
69              <td bgcolor='#0099FF'><b> 4.20</b></td>
70          <tr align='center' valign='top'>
71              <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
72              <td          bgcolor='#CCCCCC'><b>27.10.2010
16:16:38</b></td>
73              <td bgcolor='#CCCCCC'><b>20</b></td>
74              <td bgcolor='#CCCCCC'><b>56</b></td>
75              <td bgcolor='#CCCCCC'><b>47</b></td>

```

```

76      <td bgcolor='#0099FF'><b> 4.20</b></td>
77      <tr align='center' valign='top'>
78      <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
79      <td
16:03:38</b></td>          bgcolor='#CCCCCC'><b>02.11.2010
80      <td bgcolor='#CCCCCC'><b>20</b></td>
81      <td bgcolor='#CCCCCC'><b>56</b></td>
82      <td bgcolor='#CCCCCC'><b>50</b></td>
83      <td bgcolor='#0099FF'><b> 4.46</b></td>
84      <tr align='center' valign='top'>
85      <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
86      <td
16:40:50</b></td>          bgcolor='#CCCCCC'><b>02.11.2010
87      <td bgcolor='#CCCCCC'><b>20</b></td>
88      <td bgcolor='#CCCCCC'><b>56</b></td>
89      <td bgcolor='#CCCCCC'><b>46</b></td>
90      <td bgcolor='#0099FF'><b> 4.11</b></td>
91      <tr align='center' valign='top'>
92      <td bgcolor='#CCCCCC'><b>Microsoft Word</b></td>
93      <td
22:04:47</b></td>          bgcolor='#CCCCCC'><b>16.12.2010
94      <td bgcolor='#CCCCCC'><b>20</b></td>
95      <td bgcolor='#CCCCCC'><b>56</b></td>
96      <td bgcolor='#CCCCCC'><b>39</b></td>
97      <td bgcolor='#00CC99'><b> 3.48</b></td>
98      <tr align='center' valign='top'>
99      <td
Excel</b></td>          bgcolor='#CCCCCC'><b>Microsoft
100      <td
22:55:24</b></td>          bgcolor='#CCCCCC'><b>16.12.2010
101      <td bgcolor='#CCCCCC'><b>24</b></td>
102      <td bgcolor='#CCCCCC'><b>44</b></td>
103      <td bgcolor='#CCCCCC'><b>38</b></td>
104      <td bgcolor='#0099FF'><b> 4.32</b></td>
105      <tr align='center' valign='top'>
106      <td
</td>          bgcolor='#CCCCCC'><b>Microsoft Excel</b>
107      <td
22:37:43</b></td>          bgcolor='#CCCCCC'><b>28.12.2010
108      <td bgcolor='#CCCCCC'><b>24</b></td>
109      <td bgcolor='#CCCCCC'><b>44</b></td>
110      <td bgcolor='#CCCCCC'><b>37</b></td>
111      <td
</table><hr><div align='center'><a href='#usr'><b>Вернуться к
        bgcolor='#0099FF'><b> 4.20</b></td>
        список</b></a>
112      <table width='100%' border='0' cellpadding='10'>
113      <tr align='center'>
114      <td
        bgcolor='#0099FF'          align='center'
        valign='top'><b>Відмінно</b></td>

```

```

115         <td         bgcolor='#00CC99'         align='center'
valign='top'><b>Добре</b></td>
116         <td         bgcolor='#FFFF33'         align='center'
valign='top' width='25%'><b>Задовільно</b></td>
117         <td         bgcolor='#FF0000'         align='center'
valign='top' width='25%'><b>Незадовільно</b></td>
118         </tr></table></div><hr>
119     </body>
120 </html>

```

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасний підручник з JavaScript [Електрон. ресурс]. – Електрон. текст. дані. – Режим доступу: <https://uk.javascript.info/>, вільний (дата звернення 10.05.2025). – Назва з екрана.
2. Сучасний підручник з PHP [Електрон. ресурс]. – Електрон. текст. дані. – Режим доступу: <https://www.php.net/manual/uk//>, вільний (дата звернення 10.05.2025). – Назва з екрана.
3. Web-програмування. Частина 1 (frontend) : навч. посіб. / В. В. Босько, Л. В. Константинова, К. М. Марченко, О. С. Улічев. – Кропивницький : ЦНТУ, 2022. – 208 с.
4. Робсон Е. Head First. Програмування на JavaScript / Елізабет Робсон, Ерік Фрімен. – [Б. м] : Фабула, 2022. – 672 с.
5. Васильєв О. Програмування мовою PHP / О. Васильєв. – Київ : Ліра-К, 2022. – 365 с.

Електронне навчальне видання

БОЧАРОВ Борис Петрович,
ВОЄВОДИНА Марія Юріївна

**РОЗРОБКА КЛІЄНТСЬКИХ ТА СЕРВЕРНИХ ВЕБДОДАТКІВ
(FRONT-END/BACK-END)**

КОНСПЕКТ ЛЕКЦІЙ

*(для здобувачів 3 курсу першого (бакалаврського) рівня
вищої освіти денної форми навчання
зі спеціальностей 126, F6 – Інформаційні системи та технології)*

Відповідальний за випуск *М. М. Булаєнко*
Редактор *О. В. Михаленко*
Комп'ютерне верстання *Б. П. Бочаров, М. Ю. Воєводіна*

План 2021, поз. 112Л

Підп. до друку 02.06.2025. Формат 60 × 84/16.
Ум. друк. арк. 3,9.

Видавець і виготовлювач:
Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Чорноглазівська (Маршала Бажанова), 17, Харків, 61002.
Електронна адреса: office@kname.edu.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 5328 від 11.04.2017.