Срок сдачи: четверг, 25 сентября 2025, 00:00

КОНТРОЛЬНАЯ РАБОТА

по курсу «Интернет-технологии и веб-программирование»

Тема: «Разработка веб-приложения с использованием PHP и базы данных MySQL»

1. Цель работы

Целью данной контрольной работы является закрепление на практике ключевых навыков back-end-разработки: создание динамического веб-приложения, взаимодействующего с базой данных. Студент научится:

- Проектировать и создавать реляционную структуру базы данных.
- Подключаться к СУБД MySQL из PHP-скрипта.
- Реализовывать основные операции CRUD (Create, Read, Update, Delete) для управления данными.
- Организовывать код на PHP для обработки пользовательских запросов из веб-форм.
- Формировать динамические HTML-страницы на основе данных из БД.

2. Краткое описание

Студенту необходимо разработать простое, но полнофункциональное веб-приложение — **«Система управления задачами (Task Manager)»**. Приложение должно позволять пользователям просматривать список задач, добавлять новые, редактировать, помечать как выполненные и удалять существующие задачи. Все данные должны храниться в базе данных MySQL и обрабатываться серверными скриптами на PHP.

3. Ход выполнения работы

1. **Проектирование базы данных:**
    - Создать базу данных с именем `task_manager`.
    - Внутри базы создать таблицу `tasks` со следующими полями:
        - `id` (INT, PRIMARY KEY, AUTO_INCREMENT) — уникальный идентификатор задачи.
        - `title` (VARCHAR(255)) — краткое название задачи.
        - `description` (TEXT) — подробное описание задачи (может быть пустым).
        - `status` (ENUM('не выполнена', 'выполнена') или TINYINT(1)) — статус выполнения задачи.
        - `created_at` (TIMESTAMP или DATETIME) — дата и время создания задачи (значение по умолчанию: CURRENT_TIMESTAMP).
2. **Разработка структуры приложения:**
    - `index.php` — главная страница. Отображает список всех задач в виде таблицы. Для каждой задачи должны быть кнопки "Редактировать", "Удалить", "Отметить выполненной".
    - `add.php` — страница с формой для добавления новой задачи.
    - `edit.php` — страница с формой для редактирования существующей задачи (предзаполненной данными из БД).
    - `delete.php` — скрипт для удаления задачи (не имеет представления, просто обрабатывает GET/POST запрос и делает редирект).
    - `update_status.php` — скрипт для изменения статуса задачи (аналогично delete.php).
    - `config.php` — файл для подключения к базе данных (содержит логин, пароль, хост, имя БД). **Должен быть добавлен в `.gitignore` для безопасности.**
3. **Реализация функционала:**
    - **Подключение к БД:** На всех страницах, где требуется работа с данными, подключать `config.php`.
    - **Чтение (Read):** На `index.php` выполнять SQL-запрос `SELECT * FROM tasks ORDER BY created_at DESC` и выводить результат.
    - **Создание (Create):** В `add.php` принимать данные из формы (метод POST) и выполнять запрос `INSERT INTO tasks (...)`.
    - **Обновление (Update):** В `edit.php` сначала загружать данные задачи по её `id` (из GET-параметра), а затем обновлять их запросом `UPDATE tasks SET ... WHERE id = ...`. В `update_status.php` обновлять только поле `status`.
    - **Удаление (Delete):** В `delete.php` принимать `id` задачи и выполнять запрос `DELETE FROM tasks WHERE id = ...`.
4. **Валидация данных:** Обязательно проверять входящие данные (наличие `id`, длину текста, экранировать специальные символы с помощью `htmlspecialchars()` для защиты от XSS).

5. **Визуальное оформление:** Использовать базовый HTML и CSS (Bootstrap приветствуется) для создания удобного и аккуратного интерфейса.

---

4. Задания для студентов (30 вариантов)

Общее задание для всех: Выполнить разработку веб-приложения «Система управления задачами» по описанному выше плану.

Индивидуальная часть (Варианты 1-30): Каждый студент получает уникальный предмет управления. Вариант выбирается по номеру в списке журнала.

Необходимо адаптировать общую структуру таблицы `tasks` под свой вариант, переименовав её и изменив/добавив необходимые поля.

Вариант	Название предметной области (таблицы)	Дополнительные поля (кроме базовых `id`, `title`, `status`, `created_at`)
1	Список книг	`author` (VARCHAR), `year` (INT)
2	Каталог фильмов	`director` (VARCHAR), `genre` (VARCHAR)
3	Список сотрудников	`position` (VARCHAR), `salary` (DECIMAL)
4	Учёт продуктов в холодильнике	`expiry_date` (DATE), `quantity` (INT)
5	База клиентов	`email` (VARCHAR), `phone` (VARCHAR)
6	Список дел (To-Do List)	`priority` (ENUM('низкий', 'средний', 'высокий')), `due_date` (DATE)
7	Каталог товаров для интернет-магазина	`price` (DECIMAL), `category_id` (INT)
8	Учёт личных финансов	`amount` (DECIMAL), `type` (ENUM('доход', 'расход'))
9	База знаний (статьи)	`content` (TEXT), `keywords` (VARCHAR)
10	Журнал посещений врача	`doctor_name` (VARCHAR), `visit_date` (DATETIME)
11	Коллекция видеоигр	`platform` (VARCHAR), `completion_percentage` (INT)
12	Учёт автомобилей	`model` (VARCHAR), `license_plate` (VARCHAR), `year` (INT)
13	Список контактов	`phone` (VARCHAR), `address` (TEXT)
14	Учёт учебных курсов	`instructor` (VARCHAR), `hours` (INT)
15	Каталог рецептов	`cooking_time` (INT), `ingredients` (TEXT)
16	Учёт домашних животных	`type` (VARCHAR), `birth_date` (DATE)
17	База паролей (менеджер паролей)	`login` (VARCHAR), `encrypted_password` (VARCHAR) // Шифрование опционально
18	Журнал тренировок	`exercise_type` (VARCHAR), `duration` (INT)
19	Список подарков	`for_whom` (VARCHAR), `budget` (DECIMAL)
20	Учёт растений в саду	`watering_schedule` (VARCHAR), `last_watered` (DATE)
21	База идей для проектов	`category` (VARCHAR), `complexity` (ENUM('легко', 'средне', 'сложно'))
22	Каталог музыкальных альбомов	`artist` (VARCHAR), `release_year` (INT)
23	Учёт ремонтных работ	`object` (VARCHAR), `cost` (DECIMAL)
24	Список желаний (Wishlist)	`price` (DECIMAL), `link` (VARCHAR)
25	База данных сотрудников (HR)	`department` (VARCHAR), `date_of_birth` (DATE)
26	Журнал прочитанных книг	`author` (VARCHAR), `rating` (INT)
27	Учёт сериалов	`seasons` (INT), `current_episode` (INT)
28	Справочник стран	`capital` (VARCHAR), `population` (BIGINT)
29	Органайзер мероприятий	`event_date` (DATETIME), `location` (VARCHAR)
30	Учёт программного обеспечения	`version` (VARCHAR), `license_key` (VARCHAR)

Пример для варианта 1 (Список книг):

- Таблица называется `books`.
- Поля: `id`, `title`, `author`, `year`, `status` (например, "прочитана", "в процессе", "в планах"), `created_at`.

---

5. Контрольные вопросы

1. Что такое CRUD? Опишите каждую операцию и соответствующий ей SQL-запрос.
2. Как подключиться к базе данных MySQL из PHP? Перечислите основные функции (например, `mysqli_connect`).
3. В чем разница между `mysqli_fetch_assoc()` и `mysqli_fetch_array()`?
4. Для чего нужна функция `mysqli_real_escape_string()`? Что такое SQL-инъекция и как от неё защититься?
5. Что такое Prepared Statements (Подготовленные выражения) и каковы их преимущества?
6. Как передаются данные от HTML-формы к PHP-скрипту? В чем разница между методами GET и POST?
7. Как в PHP получить данные, переданные методом GET? А методом POST?
8. Для чего нужна функция `header()` в PHP? Приведите пример использования (например, редирект).

9. Что такое сессии (Sessions) в PHP? Для чего они используются и как реализуются (`session_start()`, `$_SESSION`)?
10. Как можно обеспечить безопасность паролей пользователей при хранении в БД?
11. Что такое HTTP-коды ответа? Что означают коды 200, 301, 404, 500?
12. Как организовать структуру файлов в небольшом PHP-проекте для поддержания чистоты кода?
13. Что такое PDO? Каковы его преимущества перед `mysqli`?
14. Как выполнить запрос `SELECT` с условием `WHERE` в PHP?
15. Как обработать ошибки подключения к БД или выполнения запроса?
16. Для чего используется команда `ORDER BY` в SQL? Приведите пример.
17. Как ограничить количество записей, получаемых из БД? (Оператор `LIMIT`).
18. Что такое нормализация базы данных? Какие бывают нормальные формы?
19. Опишите типы полей в MySQL: INT, VARCHAR, TEXT, DATETIME, ENUM.
20. Как создать связь между двумя таблицами в реляционной базе данных (внешний ключ - FOREIGN KEY)?
21. Что такое индекс в БД и для чего он нужен?
22. Как с помощью PHP и HTML создать динамическую выпадающий список (`<select>`), наполненный данными из БД?
23. Как загрузить файл на сервер с помощью PHP?
24. Что такое XSS (межсайтовый скриптинг) и как от него защититься на стороне PHP?
25. Для чего используется `.htaccess` файл в веб-разработке?
26. Что такое Composer и для чего он используется в PHP-проектах?
27. Как можно реализовать пагинацию (постраничную навигацию) для списка данных из БД?
28. Опишите жизненный цикл HTTP-запроса от браузера до сервера и обратно.
29. Что такое API? Как в PHP можно создать простой RESTful API?
30. Какие современные фреймворки для PHP вы знаете? (Laravel, Yii2, Symfony). В чем их основное преимущество?

Добавить ответ на задание

Состояние ответа

Номер попытки	Попытка 1.
Состояние ответа на задание	Ответы на задание еще не представлены
Состояние оценивания	Не оценено
Оставшееся время	2 дн. 10 час. осталось
Последнее изменение	-
Комментарии к ответу	› Комментарии (0)