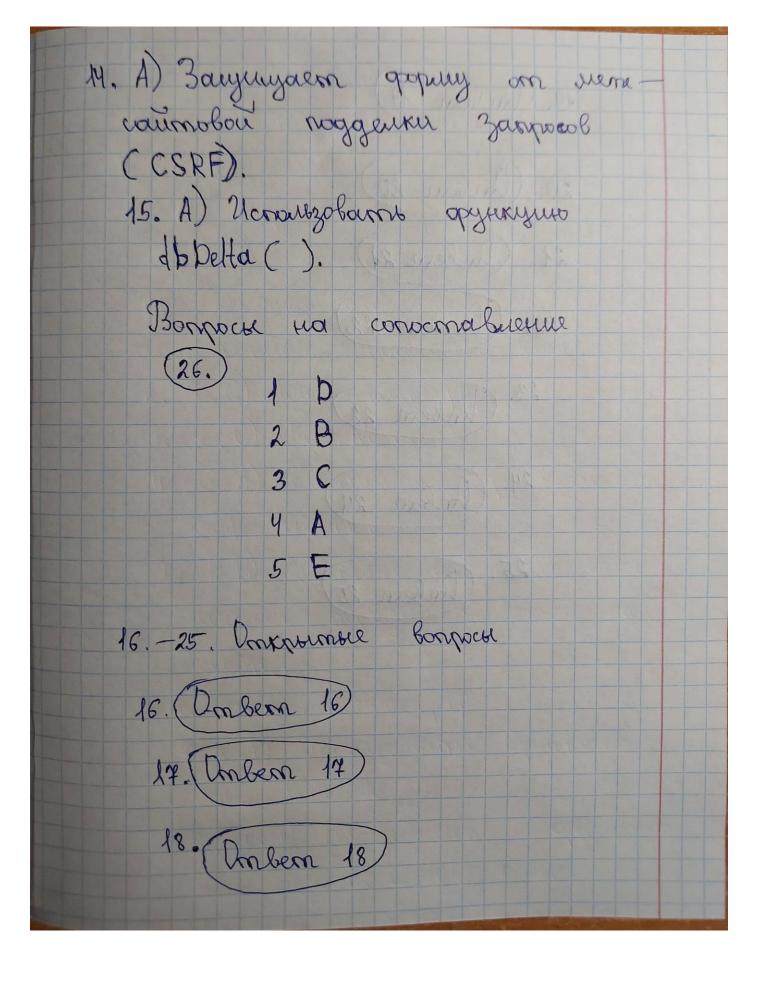
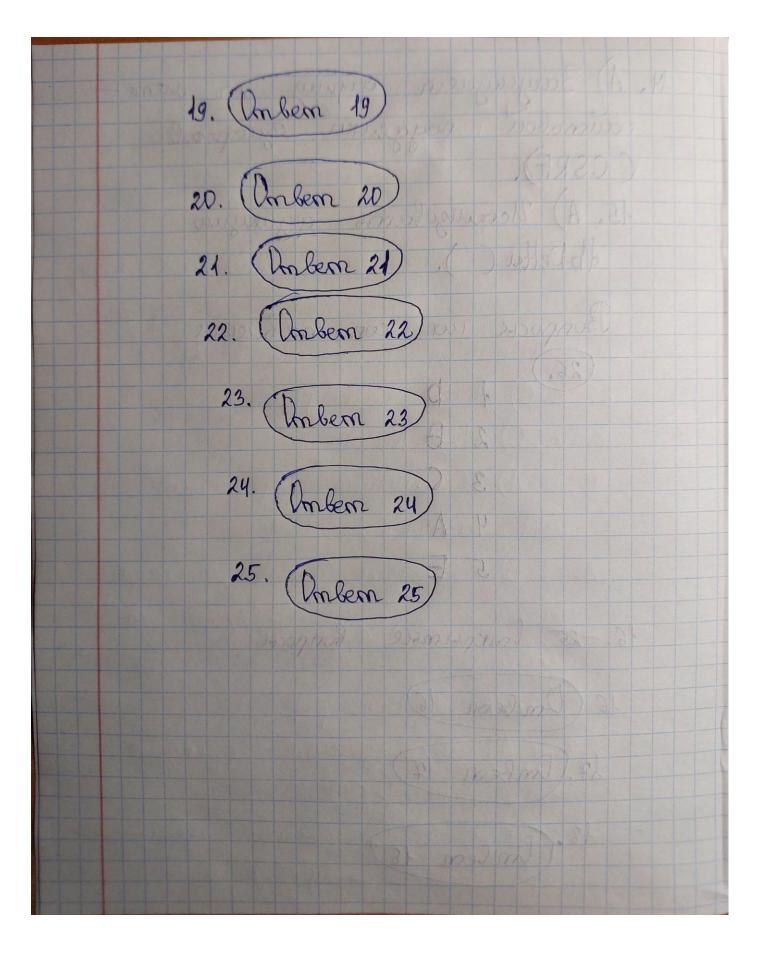
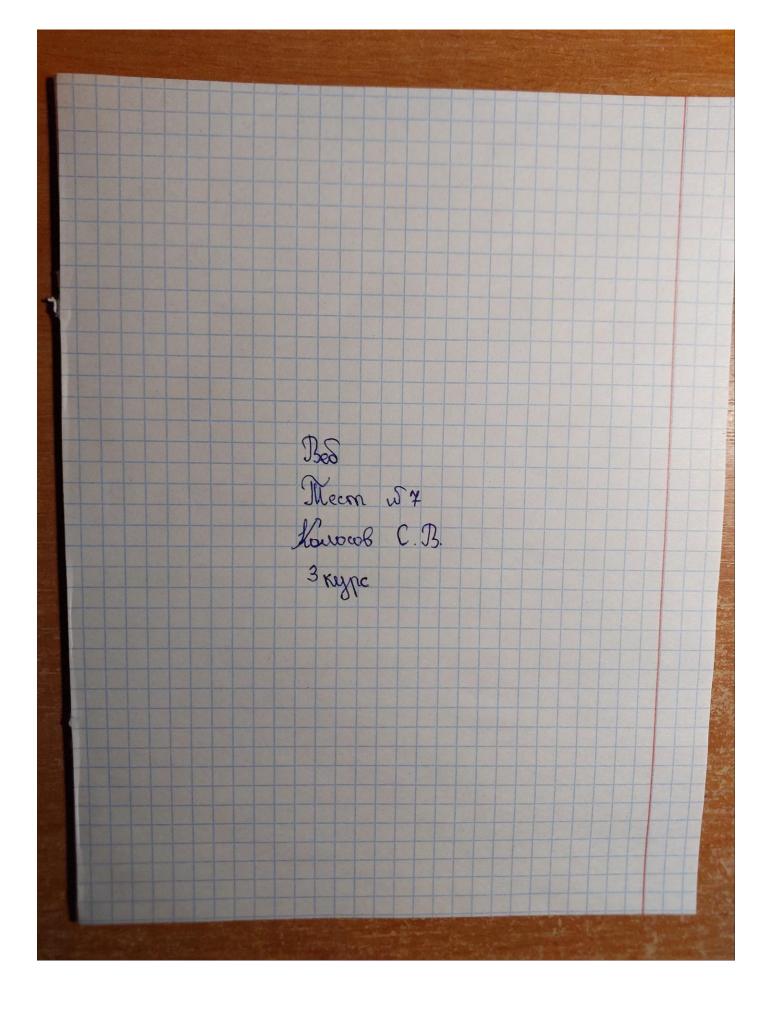
Web-noopallemenobarue
Mecon & 4
Lawoob C.B. 09.03.01 UBM -3, 3 Kype

Theen wx Memobre Koumpail, Rearnes CMS WordPress " 1-15. Bonpocte c bécopaeu 1 b. B) Cneyworlsman marka & Koge rge Montes bemalums clou kog guil mograparague robegenna Word Press 2. c. C) plugin - nome. php 3. e. C) Plugin Name ч. а. А) Спольнует функции oidd _ action () 5. p. B) Pyrkyna gwa uzwenenna garresisc repag sobribagail Ha caum; 6. a. A) functions php 7. a. A) Ucrouszobarno opyrkyuro add _ short code () 8. d. A) Ucnouszobains prejugue wp _ enqueue - style () u

Wp_enqueue_script() 9. a. A) Umalozya opyrkyuto add-menu-page () u add manu add_submenu_page() 10. A) Ino kaconpaire , cosepatwalentie & BD, Komptese Moneyo godabuarne u usurement c municipato c mornocipato populariam add_option(), update_option() u get-option (). 11. B) Imo unepopere qua bzannegenconbred e gjægnenn cephiadem u cozajarena code mbererono API; 12. А) Испаньзул федекцию add_menu_page() u add-submern _page (); 13. D) Bee boune rependant voil farmatures;







Процесс создания простого плагина в WordPress, который добавляет новый шорткод для отображения текста.

- 1. Создайте файл плагина: Создайте новый файл, например, my-shortcode-plugin.php в папке wp-content/plugins.
- 2. Добавьте заголовок плагина: В начале файла добавьте информацию о плагине.
- 3. <?php
- 4. /*
- 5. Plugin Name: My Shortcode Plugin
- 6. Description: A simple plugin to add a shortcode for displaying text.
- 7. Version: 1.0
- 8. Author: Your Name

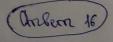
*/

- 9. **Создайте функцию для шорткода**: Определите функцию, которая будет возвращать текст.
- 10.function my_shortcode_function() {
- 11. return "Hello, this is my custom shortcode!";

12. Зарегистрируйте шорткод: Используйте функцию add_shortcode для регистрации шорткода.

add_shortcode('my_shortcode', 'my_shortcode_function');

13. **Активируйте плагин**: Перейдите в админ-панель WordPress и активируйте плагин.



Подключение внешних скриптов и стилей в плагине.

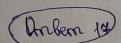
Используйте функции wp_enqueue_script и wp_enqueue_style для подключения скриптов и стилей.

function my_plugin_enqueue_scripts() {

wp_enqueue_script('my-custom-script', plugin_dir_url(__FILE__) . 'js/my-script.js', array('jquery'), '1.0', true);

wp_enqueue_style('my-custom-style', plugin_dir_url(__FILE__) . 'css/my-style.css', array(), '1.0');

add_action('wp_enqueue_scripts', 'my_plugin_enqueue_scripts');



Создание плагина, который добавляет новую страницу в админ-панель.

1. Создайте функцию для страницы настроек: 2. function my_plugin_settings_page() { add menu page(3. 'My Plugin Settings', 4. 'My Plugin', 5. 'manage options', 6. 7. 'my-plugin', 'my_plugin_settings_page content' 8. 9.); 10.} add action('admin menu', 'my plugin_settings_page'); 11. Создайте функцию для отображения содержимого страницы: 12.function my plugin_settings_page_content() { echo '<h1>My Plugin Settings</h1>'; 13.

14. echo 'Here you can configure the settings for my plugin.';

(Imbern 18)

wp_enqueue_script u wp_enqueue_style.

Эти функции используются для безопасного подключения скриптов и стилей в WordPress. Они позволяют управлять зависимостями и версиями файлов.

function my_plugin_enqueue_scripts() {

wp_enqueue_script('my-custom-script', plugin_dir_url(__FILE__) . 'js/my-script.js', array('jquery'), '1.0', true);

wp_enqueue_style('my-custom-style', plugin_dir_url(__FILE__) . 'css/my-style.css', array(), '1.0');

add_action('wp_enqueue_scripts', 'my_plugin_enqueue_scripts');

(Inben 19)

Добавление пользовательского типа записи (Custom Post Type).

```
function my_custom_post_type() {
    $args = array(
        'label' => 'Books',
        'public' => true,
        'supports' => array('title', 'editor', 'thumbnail'),
    );
    register_post_type('book', $args);
}
add_action('init', 'my_custom_post_type');
```

(Inbern 20)

Основные функции для работы с базой данных.

- global \$wpdb: Получение объекта для работы с базой данных.
- \$wpdb->get_results(): Получение результатов запроса.
- \$wpdb->insert(): Вставка данных.
- \$wpdb->update(): Обновление данных.
- \$wpdb->delete(): Удаление данных.

global \$wpdb;

\$results = \$wpdb->get_results("SELECT * FROM {\$wpdb->prefix}posts");

Inbern 21

Защита от SQL-инъекций.

- Используйте подготовленные запросы с \$wpdb->prepare().
- Экранируйте данные с помощью esc_sql().

global \$wpdb;

\$user_id = esc_sql(\$_GET['user_id']);

\$query = \$wpdb->prepare("SELECT * FROM {\$wpdb->prefix}users WHERE ID
= %d", \$user_id);

\$results = \$wpdb->get_results(\$query);

Ombem 22

Опции в WordPress.

```
Oпции используются для хранения настроек плагина.

function my_plugin_settings() {

   add_option('my_plugin_option', 'default_value');

   register_setting('my_plugin_options_group', 'my_plugin_option');
}

add_action('admin_init', 'my_plugin_settings');
```

Добавление функциональности REST API.

```
function my_plugin_rest_api() {
    register_rest_route('myplugin/v1', '/data/', array(
        'methods' => 'GET',
        'callback' => 'my_plugin_get_data',
    ));
}
add_action('rest_api_init', 'my_plugin_rest_api');

function my_plugin_get_data() {
    return new WP_REST_Response(array('data' => 'Hello, REST API!'), 200);
}
```

Vonbern 24

Хуки и фильтры в WordPress.

Хуки позволяют добавлять или изменять функциональность.

```
• Действия (Actions): add_action()
```

```
• Фильтры (Filters): add_filter()

function my_custom_action() {
  echo 'This is my custom action!';
}

add_action('init', 'my_custom_action');

function my_custom_filter($content) {
  return $content . ' This is my custom filter!';
}

add_filter('the_content', 'my_custom_filter');
```

