Library.php

```php
<?php
namespace Elementor\Modules\WpCli;

use Elementor\Api;
use Elementor\Plugin;
use Elementor\TemplateLibrary\Source_Local;

if ( ! defined( 'ABSPATH' ) ) {
    exit; // Exit if accessed directly.
}

/**
 * Elementor Page Builder cli tools.
 */
class Library extends \WP_CLI_Command {

    /**
     * Sync Elementor Library.
     *
     * [--network]
     *      Sync library in all the sites in the network.
     *
     * [--force]
     *      Force sync even if it's looks like that the library is already up to date.
     *
     * ## EXAMPLES
     *
     *  1. wp elementor library sync
     *      - This will sync the library with Elementor cloud library.
     *
     *  2. wp elementor library sync --force
     *      - This will sync the library with Elementor cloud even if it's looks like that the library is already up to date.
     *
     *  3. wp elementor library sync --network
     *      - This will sync the library with Elementor cloud library for each site in the network if needed.
     *
     * @since 2.8.0
     * @access public
     */
    public function sync( $args, $assoc_args ) {
        $network = isset( $assoc_args['network'] ) && is_multisite();

        if ( $network ) {
            $blog_ids = get_sites( [
                'fields' => 'ids',
                'number' => 0,
            ] );

            foreach ( $blog_ids as $blog_id ) {
                switch_to_blog( $blog_id );

                \WP_CLI::line( 'Site #' . $blog_id . ' - ' . get_option( 'blogname' ) );

                $this->do_sync( isset( $assoc_args['force'] ) );

                \WP_CLI::success( 'Done! - ' . get_option( 'home' ) );

                restore_current_blog();
            }
        } else {
            $this->do_sync( isset( $assoc_args['force'] ) );
            \WP_CLI::success( 'Done!' );
        }
    }

    /**
     * Import template files to the Library.
     *
     *  [--returnType]
     *      Forms of output. Possible values are 'ids', 'info'.
     *      if this parameter won't be specified, the import info will be output.
     *
     * ## EXAMPLES
     *
     *  1. wp elementor library import <file-path>
     *      - This will import a file or a zip of multiple files to the library.
     *      - file-path can be a path or url.
     *
     *  2. wp elementor library import <file-path> --returnType=info,ids
     *
     * @param $args
     * @param $assoc_args
     *
 * @since  2.8.0
 * @access public
 */
public function import( $args, $assoc_args ) {
    if ( empty( $args[0] ) ) {
        \WP_CLI::error( 'Please set file path.' );
    }

    $file = $args[0];
    $imported_items_ids = [];
    $return_type = \WP_CLI\Utils\get_flag_value( $assoc_args, 'returnType', 'info' );

    /** @var Source_Local $source */
    $source = Plugin::$instance->templates_manager->get_source( 'local' );

    if ( filter_var( $file, FILTER_VALIDATE_URL ) ) {
        $tmp_path = download_url( $file );
        if ( is_wp_error( $tmp_path ) ) {
            \WP_CLI::error( $tmp_path->get_error_message() );
        }
        $file = $tmp_path;
    }

    $imported_items = $source->import_template( basename( $file ), $file );

    if ( is_wp_error( $imported_items ) ) {
        \WP_CLI::error( $imported_items->get_error_message() );
    }

    foreach ( $imported_items as $item ) {
        $imported_items_ids[] = $item['template_id'];
    }
    $imported_items_ids = implode( ',', $imported_items_ids );

    if ( 'ids' === $return_type ) {
        \WP_CLI::line( $imported_items_ids );
    } else {
        \WP_CLI::success( count( $imported_items ) . ' item(s) has been imported.' );
    }

    if ( isset( $tmp_path ) ) {
        // Remove the temporary file, now that we're done with it.
        Plugin::$instance->uploads_manager->remove_file_or_dir( $file );
    }
}

/**
 * Import all template files from a directory.
 *
 * ## EXAMPLES
 *
 *  1. wp elementor library import-dir <file-path>
 *      - This will import all JSON files from <file-path>
 *
 * @param $args
 *
 * @since  3.4.7
 * @access public
 * @alias import-dir
 */
public function import_dir( $args ) {
    if ( empty( $args[0] ) ) {
        \WP_CLI::error( 'Please set dir path.' );
    }

    $dir = $args[0];

    if ( ! file_exists( $dir ) ) {
        \WP_CLI::error( "Dir `{$dir}` not found." );
    }

    $files = glob( $dir . '/*.json' );

    if ( empty( $files ) ) {
        \WP_CLI::error( 'Files not found.' );
    }

    /** @var Source_Local $source */
    $source = Plugin::$instance->templates_manager->get_source( 'local' );

    $succeed = [];
    $errors = [];
```

```php
    foreach ( $files as $file ) {
        $basename = basename( $file );

        if ( ! file_exists( $file ) ) {
            $errors[ $basename ] = $file . ' file not found.';
            continue;
        }

        $imported_items = $source->import_template( $basename, $file );

        if ( is_wp_error( $imported_items ) ) {
            $errors[ $basename ] = $imported_items->get_error_message();
        } else {
            $succeed[ $basename ] = true;
        }
    }

    $succeed_message = count( $succeed ) . ' item(s) has been imported.';

    if ( ! empty( $errors ) ) {
        $error_message = var_export( $errors, 1 );
        if ( ! empty( $succeed ) ) {
            $error_message = $succeed_message . ' ' . count( $errors ) . ' has errors: ' . $error_message;
        }
        \WP_CLI::error( $error_message );
    }

    \WP_CLI::success( $succeed_message );
}

/**
 * Connect site to Elementor Library.
 * (Network is not supported)
 *
 * --user
 *      The user to connect <id|login|email>
 *
 * --token
 *      A connect token from Elementor Account Dashboard.
 *
 * ## EXAMPLES
 *
 *  1. wp elementor library connect --user=admin --token=<connect-cli-token>
 *      - This will connect the admin to Elementor library.
 *
 * @param $args
 * @param $assoc_args
 *
 * @since  2.8.0
 * @access public
 */
public function connect( $args, $assoc_args ) {
    if ( ! get_current_user_id() ) {
        \WP_CLI::error( 'Please set user to connect (--user=<id|login|email>).' );
    }

    if ( empty( $assoc_args['token'] ) ) {
        \WP_CLI::error( 'Please set connect token.' );
    }

    $_REQUEST['mode'] = 'cli';
    $_REQUEST['token'] = $assoc_args['token'];

    $app = $this->get_library_app();

    $app->set_auth_mode( 'cli' );

    $app->action_authorize();

    $app->action_get_token();
}

/**
 * Disconnect site from Elementor Library.
 *
 * --user
 *      The user to disconnect <id|login|email>
 *
 * ## EXAMPLES
 *
 *  1. wp elementor library disconnect --user=admin
 *      - This will disconnect the admin from Elementor library.
 *
```

```
     * @param $args
     * @param $assoc_args
     *
     * @since  2.8.0
     * @access public
     */
    public function disconnect() {
        if ( ! get_current_user_id() ) {
            \WP_CLI::error( 'Please set user to connect (--user=<id|login|email>).' );
        }

        $_REQUEST['mode'] = 'cli';

        $this->get_library_app()->action_disconnect();
    }

    private function do_sync() {
        $data = Api::get_library_data( true );

        if ( empty( $data ) ) {
            \WP_CLI::error( 'Cannot sync library.' );
        }
    }

    /**
     * @return \Elementor\Core\Common\Modules\Connect\Apps\Library
     */
    private function get_library_app() {
        $connect = Plugin::$instance->common->get_component( 'connect' );
        $app = $connect->get_app( 'library' );
        // Before init.
        if ( ! $app ) {
            $connect->init();
            $app = $connect->get_app( 'library' );
        }

        return $app;
    }
}
```