

URL Health Monitor — A minimalist WordPress plugin

📄 [Claude Opus 4.5]

Below is a complete, single-file WordPress plugin that does exactly what you described. It's deliberately minimal while still being production-ready.

Features at a glance

- **Simple backend UI** — a single textarea where you enter URLs (one per line)
 - **WP-Cron powered** — checks every hour by default
 - **Lightweight storage** — results saved in `wp_options`, auto-pruned after 365 days
 - **Frontend shortcode** — `[url_health_monitor]` outputs a clean, responsive status table
 - **Zero dependencies** — pure WordPress APIs, no external libraries
-

The plugin code

Create a file called `url-health-monitor.php` inside `wp-content/plugins/url-health-monitor/` and paste the following:

```
<?php
/**
 * Plugin Name: URL Health Monitor
 * Description: A minimalist plugin to monitor URL availability and display results via
shortcode.
 * Version: 1.0.0
 * Author: Your Name
 * License: GPL-2.0+
 */
```

```

if ( ! defined( 'ABSPATH' ) ) {
    exit;
}

class URL_Health_Monitor {

    const OPTION_URLS      = 'uhm_urls';
    const OPTION_RESULTS  = 'uhm_results';
    const CRON_HOOK       = 'uhm_check_urls';
    const RETENTION_DAYS = 365;

    public function __construct() {
        // Admin hooks
        add_action( 'admin_menu', [ $this, 'add_admin_menu' ] );
        add_action( 'admin_init', [ $this, 'register_settings' ] );

        // Cron
        add_action( self::CRON_HOOK, [ $this, 'check_all_urls' ] );
        add_filter( 'cron_schedules', [ $this, 'add_cron_interval' ] );

        // Shortcode
        add_shortcode( 'url_health_monitor', [ $this, 'render_shortcode' ] );

        // Activation / Deactivation
        register_activation_hook( __FILE__, [ $this, 'activate' ] );
        register_deactivation_hook( __FILE__, [ $this, 'deactivate' ] );
    }

    /* -----
       Activation & Deactivation
    ----- */

    public function activate() {
        if ( ! wp_next_scheduled( self::CRON_HOOK ) ) {
            wp_schedule_event( time(), 'hourly', self::CRON_HOOK );
        }
    }

    public function deactivate() {

```

```

        wp_clear_scheduled_hook( self::CRON_HOOK );
    }

    public function add_cron_interval( $schedules ) {
        // You can change 'hourly' above to 'uhm_fifteen_minutes' and uncomment below for
        // faster checks.
        // $schedules['uhm_fifteen_minutes'] = [
        //     'interval' => 900,
        //     'display' => __( 'Every 15 Minutes' ),
        // ];
        return $schedules;
    }

    /* -----
    Admin Menu & Settings
    ----- */

    public function add_admin_menu() {
        add_options_page(
            'URL Health Monitor',
            'URL Health Monitor',
            'manage_options',
            'url-health-monitor',
            [ $this, 'render_admin_page' ]
        );
    }

    public function register_settings() {
        register_setting( 'uhm_settings_group', self::OPTION_URLS, [
            'type' => 'string',
            'sanitize_callback' => [ $this, 'sanitize_urls' ],
            'default' => '',
        ] );
    }

    public function sanitize_urls( $input ) {
        $lines = explode( "\n", $input );
        $clean = [];
        foreach ( $lines as $line ) {
            $url = esc_url_raw( trim( $line ) );

```

```

        if ( $url ) {
            $clean[] = $url;
        }
    }
    return implode( "\n", $clean );
}

public function render_admin_page() {
    if ( ! current_user_can( 'manage_options' ) ) {
        return;
    }
    $urls = get_option( self::OPTION_URLS, '' );
    ?>
    <div class="wrap" style="max-width:720px;">
        <h1>URL Health Monitor</h1>
        <form method="post" action="options.php">
            <?php settings_fields( 'uhm_settings_group' ); ?>
            <p>Enter the URLs you want to monitor, <strong>one per line</strong>:</p>
            <textarea name="<?php echo esc_attr( self::OPTION_URLS ); ?>" rows="10"
style="width:100%;font-family:monospace;"><?php echo esc_textarea( $urls ); ?></textarea>
            <?php submit_button( 'Save URLs' ); ?>
        </form>

        <hr>
        <h2>Current status</h2>
        <?php $this->render_admin_status_table(); ?>

        <hr>
        <h2>Shortcode</h2>
        <p>Use <code>[url_health_monitor]</code> to display the status table on the
frontend.</p>
    </div>
    <?php
}

private function render_admin_status_table() {
    $results = $this->get_results();
    if ( empty( $results ) ) {
        echo '<p><em>No data yet. Results will appear after the first scheduled check
(runs hourly).</em></p>';
    }
}

```

```

        echo '<form method="post">';
        wp_nonce_field( 'uhm_manual_check', 'uhm_nonce' );
        echo '<button type="submit" name="uhm_run_now" class="button">Run check
now</button>';
        echo '</form>';

        if ( isset( $_POST['uhm_run_now'] ) && check_admin_referer( 'uhm_manual_check',
'uhm_nonce' ) ) {
            $this->check_all_urls();
            echo '<meta http-equiv="refresh" content="0">';
        }
        return;
    }

    echo '<form method="post">';
    wp_nonce_field( 'uhm_manual_check', 'uhm_nonce' );
    echo '<button type="submit" name="uhm_run_now" class="button" style="margin-
bottom:12px;">Run check now</button>';
    echo '</form>';

    if ( isset( $_POST['uhm_run_now'] ) && check_admin_referer( 'uhm_manual_check',
'uhm_nonce' ) ) {
        $this->check_all_urls();
        echo '<meta http-equiv="refresh" content="0">';
    }

    $this->render_status_table( $results, false );
}

/* -----
   URL Checking Logic
   ----- */

public function check_all_urls() {
    $raw = get_option( self::OPTION_URLS, '' );
    if ( empty( $raw ) ) {
        return;
    }

    $urls = array_filter( array_map( 'trim', explode( "\n", $raw ) ) );

```

```

$results = $this->get_results();
$now      = current_time( 'timestamp' );

foreach ( $urls as $url ) {
    $response = wp_remote_head( $url, [
        'timeout'      => 15,
        'redirection' => 5,
        'sslverify'   => false,
    ] );

    $status_code = 0;
    $is_up       = false;

    if ( is_wp_error( $response ) ) {
        $status_code = 0;
    } else {
        $status_code = wp_remote_retrieve_response_code( $response );
        $is_up       = ( $status_code >= 200 && $status_code < 400 );
    }

    if ( ! isset( $results[ $url ] ) ) {
        $results[ $url ] = [ 'history' => [] ];
    }

    $results[ $url ]['last_checked'] = $now;
    $results[ $url ]['status_code']  = $status_code;
    $results[ $url ]['is_up']        = $is_up;
    $results[ $url ]['history'][]    = [
        'time'   => $now,
        'status' => $status_code,
        'is_up'  => $is_up,
    ];
}

// Prune old history & orphan URLs
$results = $this->prune_results( $results, $urls );

update_option( self::OPTION_RESULTS, $results );
}

```

```

private function prune_results( $results, $current_urls ) {
    $cutoff = current_time( 'timestamp' ) - ( self::RETENTION_DAYS * DAY_IN_SECONDS );

    foreach ( $results as $url => &$data ) {
        // Remove URLs no longer monitored
        if ( ! in_array( $url, $current_urls, true ) ) {
            unset( $results[ $url ] );
            continue;
        }

        // Remove history entries older than retention period
        $data['history'] = array_filter( $data['history'], function ( $entry ) use (
$cutoff ) {
            return $entry['time'] >= $cutoff;
        } );
        $data['history'] = array_values( $data['history'] ); // reindex
    }

    return $results;
}

private function get_results() {
    return get_option( self::OPTION_RESULTS, [] );
}

/* -----
Frontend Shortcode
----- */

public function render_shortcode( $atts ) {
    $results = $this->get_results();
    if ( empty( $results ) ) {
        return '<p>No monitoring data available yet.</p>';
    }

    ob_start();
    $this->render_status_table( $results, true );
    return ob_get_clean();
}

```

```

private function render_status_table( $results, $is_frontend ) {
    $border_color = '#e0e0e0';
    $up_color     = '#2e7d32';
    $down_color   = '#c62828';
    ?>
    <style>
        .uhm-table { width:100%; border-collapse:collapse; font-family: -apple-system,
BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen, Ubuntu, sans-serif; font-size:14px; }
        .uhm-table th, .uhm-table td { padding:10px 12px; text-align:left; border-
bottom:1px solid <?php echo $border_color; ?>; }
        .uhm-table th { background:#f7f7f7; font-weight:600; }
        .uhm-table tr:last-child td { border-bottom:none; }
        .uhm-status { display:inline-block; padding:3px 10px; border-radius:4px; font-
weight:500; font-size:12px; }
        .uhm-status.up { background:#e8f5e9; color:<?php echo $up_color; ?>; }
        .uhm-status.down { background:#ffebee; color:<?php echo $down_color; ?>; }
        .uhm-url { word-break:break-all; }
        .uhm-time { color:#666; font-size:12px; }
        .uhm-uptime { font-size:12px; color:#555; }
    </style>
    <table class="uhm-table">
        <thead>
            <tr>
                <th>URL</th>
                <th>Status</th>
                <th>Uptime (<?php echo self::RETENTION_DAYS; ?>d)</th>
                <th>Last checked</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ( $results as $url => $data ) :
                $is_up      = ! empty( $data['is_up'] );
                $status_code = isset( $data['status_code'] ) ? $data['status_code'] : '-';
                $last_time   = isset( $data['last_checked'] ) ? human_time_diff(
$data['last_checked'], current_time( 'timestamp' ) ) . ' ago' : '-';

                // Calculate uptime %
                $history = isset( $data['history'] ) ? $data['history'] : [];
                $total   = count( $history );
                $up_count = 0;

```

```

        foreach ( $history as $h ) {
            if ( ! empty( $h['is_up'] ) ) {
                $up_count++;
            }
        }
        $uptime = $total > 0 ? round( ( $up_count / $total ) * 100, 1 ) : 0;
        ?>
        <tr>
            <td class="uhm-url"><?php echo esc_html( $url ); ?></td>
            <td>
                <span class="uhm-status <?php echo $is_up ? 'up' : 'down'; ?>">
                    <?php echo $is_up ? '✓ Online' : '* Offline'; ?>
                </span>
                <span style="margin-left:6px;color:#888;font-size:12px;"><?php echo
esc_html( $status_code ); ?></span>
            </td>
            <td class="uhm-uptime"><?php echo esc_html( $uptime ); ?>%</td>
            <td class="uhm-time"><?php echo esc_html( $last_time ); ?></td>
        </tr>
        <?php endforeach; ?>
    </tbody>
</table>
<?php
}
}

new URL_Health_Monitor();

```

How to install

1. Create the folder `wp-content/plugins/url-health-monitor/`
2. Save the code above as `url-health-monitor.php` inside that folder
3. Activate the plugin via **Plugins → Installed Plugins**

How to use

1. Configure URLs

Go to **Settings → URL Health Monitor**. Enter the URLs you want to watch (one per line) and click *Save URLs*.

2. Wait or trigger manually

The plugin checks URLs every hour via WP-Cron. You can also click *Run check now* in the admin screen to trigger an immediate check.

3. Display on the frontend

Insert the shortcode anywhere:

```
[url_health_monitor]
```

It will output a clean status table showing each URL's current state, HTTP status code, uptime percentage, and last-checked time.

Notes on WP-Cron ☐

WordPress cron is *visitor-triggered* by default — it only fires when someone visits your site. If your site has low traffic, consider adding a real server cron job:

```
*/15 * * * * curl -s https://yoursite.com/wp-cron.php?doing_wp_cron > /dev/null 2>&1
```

Then disable the built-in trigger by adding this to `wp-config.php`:

```
define( 'DISABLE_WP_CRON', true );
```

Customisation ideas ☐☐

Want to change...	Where to look
Check interval	Change <code>'hourly'</code> in <code>activate()</code> to <code>'twicedaily'</code> , <code>'daily'</code> , or register your own schedule via <code>add_cron_interval()</code>
Retention period	Adjust <code>RETENTION_DAYS</code> constant
Styling	Modify the <code><style></code> block inside <code>render_status_table()</code>
Timeout / SSL verification	Tweak the <code>wp_remote_head()</code> args in <code>check_all_urls()</code>

That's it — a clean, minimal URL health monitor in under 300 lines of code. ☐☐

Updated 2026-05-08 21:56:36 UTC by art10m