OŚWIADCZENIE O ZGODNOŚCI Z WYTYCZNYMI WCAG 2.1

Wykonawca:

BIURO USŁUG KOMPUTEROWYCH "SOFTRES" SPÓŁKA Z OGRANICZONĄ ODPOWIEDZIALNOŚCIĄ
KRS: 0000083971
NIP: 8130335217
REGON: 690037603
Adres siedziby: ul. Zaciszna 44, 35-326 Rzeszów, Polska

Odbiorca:

Urząd Miasta Krosna
Adres: 38-400 Krosno ul. Lwowska 28a

Oświadczenie

Ja, niżej podpisany, reprezentując firmę BIURO USŁUG KOMPUTEROWYCH "SOFTRES" Sp. z o.o., oświadczam, że portal internetowy „eBOM" dla Urzędu Miasta Krosna, dostępny pod adresem: https://eurzad.umkrosno.pl został wykonany zgodnie z wytycznymi Web Content Accessibility Guidelines (WCAG) 2.1 na poziomie AA, zgodnie z wymaganiami określonymi w:

- Rozporządzeniu Ministra Cyfryzacji z dnia 12 kwietnia 2019 r. w sprawie dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych (Dz.U. 2019 poz. 848),
- oraz zgodnie z normą EN 301 549.

Zakres spełnienia wymagań WCAG 2.1

W trakcie projektowania i wdrożenia serwisu zapewniono zgodność w następujących obszarach:

1. Postrzegalność (Perceivable)

- zapewniono alternatywne opisy dla obrazów, ikon i elementów graficznych,
- wszystkie treści posiadają odpowiedni kontrast między tekstem a tłem,
- strona jest w pełni responsywna i umożliwia powiększenie do 200% bez utraty czytelności,
- multimedia nie uruchamiają się automatycznie i nie powodują rozproszenia uwagi użytkownika.

2. Funkcjonalność (Operable)

- wszystkie funkcje strony są dostępne przy użyciu klawiatury,
- nawigacja jest logiczna, spójna i przewidywalna,
- linki i przyciski są jednoznacznie opisane,
- nie występują błyski ani animacje mogące powodować dyskomfort użytkownika.

3. Zrozumiałość (Understandable)

- struktura nagłówków i treści jest logiczna i hierarchiczna,
- formularze zawierają prawidłowe etykiety, pola i komunikaty błędów,
- język strony został zdefiniowany w znaczniku HTML.

4. Solidność (Robust)

- kod HTML i ARIA są zgodne ze standardami W3C,
- strona jest poprawnie interpretowana przez czytniki ekranu,
- zastosowano poprawną semantykę elementów interfejsu.

Metody weryfikacji dostępności

Ocena zgodności została przeprowadzona przy użyciu testów automatycznych i manualnych, obejmujących:

- analizę kodu HTML, struktur semantycznych i znaczników ARIA,
- testy z wykorzystaniem narzędzi automatycznych:
    - WAVE WebAIM – wynik: *0 błędów (Errors)*, *0 błędów kontrastu (Contrast Errors)*, AIM Score: 9.9/10,
    - Lighthouse (Chrome DevTools) – wynik: *Accessibility Score: 93/100*,
- testy manualne przeprowadzone przez wykonawcę w zakresie:
    - obsługi strony wyłącznie za pomocą klawiatury,
    - poprawności odczytu treści przez czytniki ekranu,
    - zgodności struktury nagłówków i etykiet formularzy,
    - poprawności działania elementów interaktywnych (menu, przyciski, linki).

Testy potwierdzają, że strona spełnia wymagania WCAG 2.1 na poziomie AA i jest zgodna z aktualnymi standardami dostępności cyfrowej.

Załączniki:

1. Raport z narzędzia Lighthouse – plik „audyt_lighthouse.pdf”,
2. Zrzut ekranu z narzędzia WAVE – plik „audyt_WAVE.pdf”.

Niniejsze oświadczenie zostało sporządzone na podstawie wyników testów automatycznych i manualnych przeprowadzonych w dniu 2 lutego 2026 roku. Dotyczy ono stanu faktycznego strony internetowej na ten dzień.

Data: 2 luty 2026 r.
Miejsce: Rzeszów

Dokument podpisany przez Tomasz Szczęsny
Data: 2026.02.02 17:35:17 CET

Podpis / pieczęć wykonawcy:

.....

Załącznik 1.

93

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

NAMES AND LABELS

▲ Select elements do not have associated label elements.

Form elements without effective labels can create frustrating experiences for screen reader users. Learn more about the `select` element.

Failing Elements
select#listLangComboBox.form-select

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

NAVIGATION

▲ Heading elements are not in a sequentially-descending order

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. Learn more about heading order.

Failing Elements	
	h5.menu-name

These are opportunities to improve keyboard navigation in your application.

BEST PRACTICES

▲ Document does not have a main landmark.

One main landmark helps screen reader users navigate a web page. Learn more about landmarks.

Failing Elements	
	html.dxChrome.dxWindowsPlatform.dxWebKitFamily.dxBrowserVersion-144

These items highlight common accessibility best practices.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

- Interactive controls are keyboard focusable

  Custom interactive controls are keyboard focusable and display a focus indicator. Learn how to make custom controls focusable. Unscored

- Interactive elements indicate their purpose and state

  Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. Learn how to decorate interactive elements with affordance hints. Unscored

- The page has a logical tab order

  Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more about logical tab ordering. Unscored

- Visual order on the page follows DOM order

  DOM order matches the visual order, improving navigation for assistive technology. Learn more about DOM and visual ordering. Unscored

- User focus is not accidentally trapped in a region

  A user can tab into and out of any control or region without accidentally trapping their focus. Learn how to avoid focus traps. Unscored

- The user's focus is directed to new content added to the page

If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn how to direct focus to new content. Unscored

- HTML5 landmark elements are used to improve navigation

Landmark elements (`<main>`, `<nav>`, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more about landmark elements. Unscored

- Offscreen content is hidden from assistive technology

Offscreen content is hidden with display: none or aria-hidden=true. Learn how to properly hide offscreen content. Unscored

- Custom controls have associated labels

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more about custom controls and labels. Unscored

- Custom controls have ARIA roles

Custom interactive controls have appropriate ARIA roles. Learn how to add roles to custom controls. Unscored

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (25) Hide

- `[aria-*]` attributes match their roles

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. Learn how to match ARIA attributes to their roles.

- `[aria-hidden="true"]` is not present on the document `<body>`

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document `<body>`. Learn how `aria-hidden` affects the document body.

- `[role]`s have all required `[aria-*]` attributes

Some ARIA roles have required attributes that describe the state of the element to screen readers. Learn more about roles and required attributes.

- Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children.

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. Learn more about roles and required children elements.

`[role]`s are contained by their required parent element

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. Learn more about ARIA roles and required parent element.

`[role]` values are valid

ARIA roles must have valid values in order to perform their intended accessibility functions. Learn more about valid ARIA roles.

`[aria-*]` attributes have valid values

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. Learn more about valid values for ARIA attributes.

`[aria-*]` attributes are valid and not misspelled

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. Learn more about valid ARIA attributes.

Buttons have an accessible name

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. Learn how to make buttons more accessible.

Image elements have `[alt]` attributes

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. Learn more about the `alt` attribute.

`[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. Learn more about the viewport meta tag.

ARIA attributes are used as specified for the element's role

Some ARIA attributes are only allowed on an element under certain conditions. Learn more about conditional ARIA attributes.

`[aria-hidden="true"]` elements do not contain focusable descendents

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. Learn how `aria-hidden` affects focusable elements.

Elements use only permitted ARIA attributes

Using ARIA attributes in roles where they are prohibited can mean that important information is not communicated to users of assistive technologies. Learn more about prohibited ARIA roles.

Background and foreground colors have a sufficient contrast ratio

Low-contrast text is difficult or impossible for many users to read. Learn how to provide sufficient color contrast.

Document has a `<title>` element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more about document titles.

`<html>` element has a `[lang]` attribute

If a page doesn't specify a `lang` attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute.

`<html>` element has a valid value for its `[lang]` attribute

Specifying a valid BCP 47 language helps screen readers announce text properly. Learn how to use the `lang` attribute.

Links are distinguishable without relying on color.

Low-contrast text is difficult or impossible for many users to read. Link text that is discernible improves the experience for users with low vision. Learn how to make links distinguishable.

Links have a discernible name

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. Learn how to make links accessible.

Lists contain only `<li>` elements and script supporting elements (`<script>` and `<template>`).

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more about proper list structure.

List items (`<li>`) are contained within `<ul>`, `<ol>` or `<menu>` parent elements

Screen readers require list items (`<li>`) to be contained within a parent `<ul>`, `<ol>` or `<menu>` to be announced properly. Learn more about proper list structure.

No element has a `[tabindex]` value greater than 0

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. Learn more about the `tabindex` attribute.

Touch targets have sufficient size and spacing.

Touch targets with sufficient size and spacing help users who may have difficulty targeting small controls to activate the targets. Learn more about touch targets.

Deprecated ARIA roles were not used

Deprecated ARIA roles may not be processed correctly by assistive technology. Learn more about deprecated ARIA roles.

NOT APPLICABLE (32)

Hide

`[accesskey]` values are unique

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. Learn more about access keys. Unscored

`button`, `link`, and `menuitem` elements have accessible names

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to make command elements more accessible. Unscored

Elements with `role="dialog"` or `role="alertdialog"` have accessible names.

ARIA dialog elements without accessible names may prevent screen readers users from discerning the purpose of these elements. Learn how to make ARIA dialog elements more accessible. Unscored

ARIA input fields have accessible names

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about input field labels. Unscored

ARIA `meter` elements have accessible names

When a meter element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to name `meter` elements. Unscored

- ARIA `progressbar` elements have accessible names

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to label `progressbar` elements. Unscored

- Elements with the `role=text` attribute do not have focusable descendents.

Adding `role=text` around a text node split by markup enables VoiceOver to treat it as one phrase, but the element's focusable descendents will not be announced. Learn more about the `role=text` attribute. Unscored

- ARIA toggle fields have accessible names

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about toggle fields. Unscored

- ARIA `tooltip` elements have accessible names

When a tooltip element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn how to name `tooltip` elements. Unscored

- ARIA `treeitem` elements have accessible names

When a `treeitem` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more about labeling `treeitem` elements. Unscored

- The page contains a heading, skip link, or landmark region

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. Learn more about bypass blocks. Unscored

- `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements.

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. Learn how to structure definition lists correctly. Unscored

- Definition list items are wrapped in `<dl>` elements

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. Learn how to structure definition lists correctly. Unscored

- ARIA IDs are unique

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. Learn how to fix duplicate ARIA IDs. Unscored

- No form fields have multiple labels

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. Learn how to use form labels. Unscored

- `<frame>` or `<iframe>` elements have a title

Screen reader users rely on frame titles to describe the contents of frames. Learn more about frame titles. Unscored

- `<html>` element has an `[xml:lang]` attribute with the same base language as the `[lang]` attribute.

If the webpage does not specify a consistent language, then the screen reader might not announce the page's text correctly. Learn more about the `lang` attribute. Unscored

- Input buttons have discernible text.

Adding discernable and accessible text to input buttons may help screen reader users understand the purpose of the input button. Learn more about input buttons. Unscored

- `<input type="image">` elements have `[alt]` text

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. Learn about input image alt text. Unscored

- Form elements have associated labels

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. Learn more about form element labels. Unscored

- The document does not use `<meta http-equiv="refresh">`

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. Learn more about the refresh meta tag. Unscored

- `<object>` elements have alternate text

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. Learn more about alt text for `object` elements. Unscored

- Skip links are focusable.

Including a skip link can help users skip to the main content to save time. Learn more about skip links. Unscored

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table.

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. Learn more about the `headers` attribute. Unscored

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe.

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. Learn more about table headers. Unscored

- `[lang]` attributes have a valid value

Specifying a valid BCP 47 language on elements helps ensure that text is pronounced correctly by a screen reader. Learn how to use the `lang` attribute. Unscored

- `<video>` elements contain a `<track>` element with `[kind="captions"]`

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. Learn more about video captions. Unscored

- Tables have different content in the summary attribute and `<caption>`.

The summary attribute should describe the table structure, while `<caption>` should have the onscreen title. Accurate table mark-up helps users of screen readers. Learn more about summary and caption. Unscored

- All heading elements contain content.

A heading with no content or inaccessible text prevent screen reader users from accessing information on the page's structure. Learn more about headings. Unscored

- Uses ARIA roles only on compatible elements

Many HTML elements can only be assigned certain ARIA roles. Using ARIA roles where they are not allowed can interfere with the accessibility of the web page. Learn more about ARIA roles. Unscored

- Image elements do not have `[alt]` attributes that are redundant text.

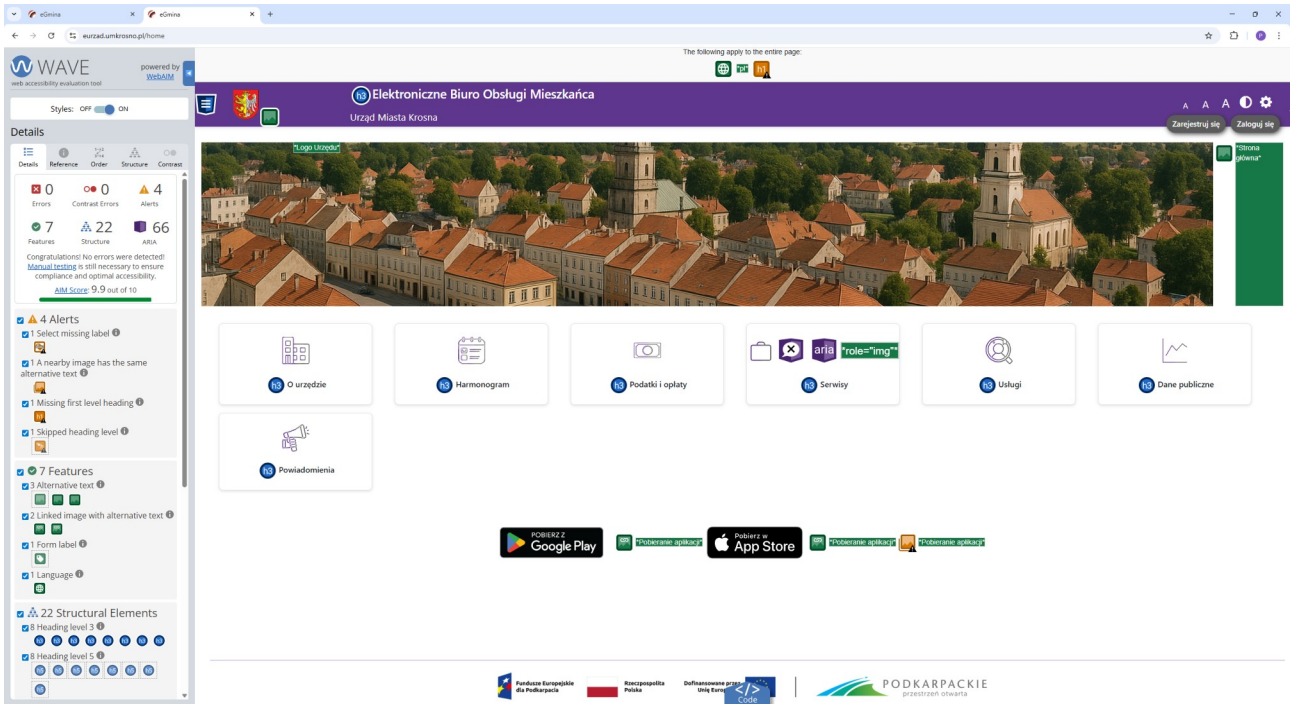
Informative elements should aim for short, descriptive alternative text. Alternative text that is exactly the same as the text adjacent to the link or image is potentially confusing for screen reader users, because the text will be read twice. Learn more about the `alt` attribute. Unscored

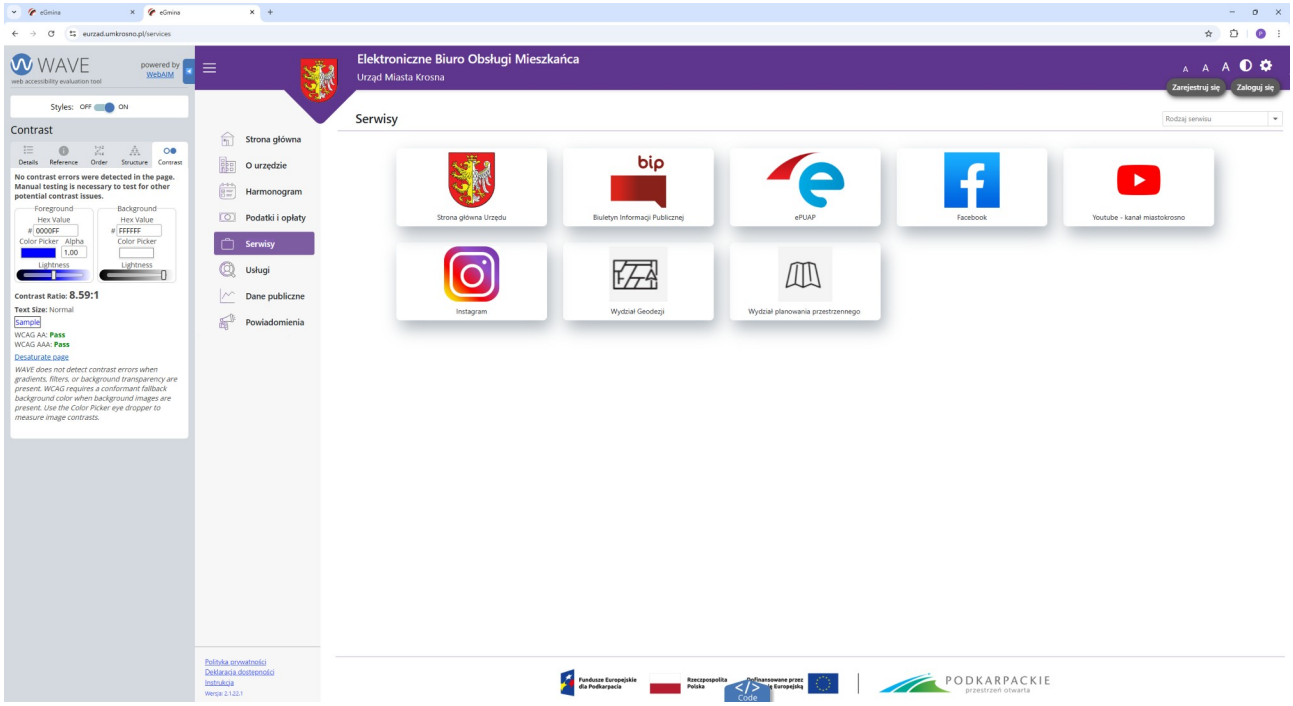
Identical links have the same purpose.

Links with the same destination should have the same description, to help users understand the link's purpose and decide whether to follow it. Learn more about identical links. Unscored

Captured at Feb 2, 2026, 4:28 PM GMT+1	Emulated Desktop with Lighthouse 13.0.1	Single page session
Initial page load	Custom throttling	Using Chromium 144.0.0.0 with devtools

Załącznik 2.


WAVE
Details
Elektroniczne Biuro Obsługi Mieszkańca
Urząd Miasta Krosna
O urzędzie
Harmonogram
Podatki i opłaty
Serwisy
Usługi
Dane publiczne
Powiadomienia
PODKARPACKIE


WAVE
Contrast
Elektroniczne Biuro Obsługi Mieszkańca
Urząd Miasta Krosna
Serwisy
Strona główna
O urzędzie
Harmonogram
Podatki i opłaty
Usługi
Dane publiczne
Powiadomienia
Strona główna Urzędu
Biuletyn Informacji Publicznej
ePUAP
Facebook
Youtube - kanał miastokrosno
Instagram
Wydział Geodezji
Wydział planowania przestrzennego
PODKARPACKIE