

---

# Prohlášení o užití generativní AI

---

## Vážené studentky, vážení studenti,

tento formulář Vás provede tvorbou *Prohlášení o užití umělé inteligence*. V tomto prohlášení shrnete, zda byla při tvorbě práce užitá umělá inteligence a popíšete, jakým způsobem.

Univerzita Karlova podporuje využívání nástrojů generativní umělé inteligence studujícími za předpokladu, že je využití AI transparentní a v souladu s právním řádem ČR, předpisy Univerzity Karlovy či jejich součástí.

Níže si můžete vygenerovat základní podobu Prohlášení. Doporučujeme zkontrolovat, zda vaše fakulta nepožaduje jinou formu prohlášení, případně konzultovat se svým školitelem či školitelkou.

```
<div id="ai-declaration-widjet"> <h2 id="ai-title">Prohlášení o využití AI</h2> <p id="ai-js-warning"> Tento formulář vyžaduje povolený JavaScript. Pokud po kliknutí na tlačítko nevznikne prohlášení, redakční systém pravděpodobně odstraňuje skripty. V takovém případě použijte tento formulář jen jako vodítko a výsledné prohlášení zapište ručně podle textů níže. </p> <input type="hidden" id="ai-lang" name="ai-lang" value="cs"> <br> <fieldset> <legend id="ai-legend-used">1. Použití AI</legend> <p id="ai-question-used">Použil/a jste při tvorbě této práce nějaký nástroj umělé inteligence?</p> <p> <label><input type="radio" name="ai-used" value="no"> <span id="ai-label-used-no">Ne, nepoužil/a jsem žádné nástroje AI.</span></label><br> <label><input type="radio" name="ai-used" value="yes" checked> <span id="ai-label-used-yes">Ano, použil/a jsem alespoň jeden nástroj AI.</span></label> </p> </fieldset> <br> <fieldset id="ai-tool-section"> <legend id="ai-legend-tool">2. Použitý nástroj</legend> <p> <label id="ai-label-tool" for="ai-tool">Nástroj AI</label><br> <select id="ai-tool" name="ai-tool"> <option value="Microsoft Copilot">Microsoft Copilot</option> <option value="ChatGPT">ChatGPT</option> <option value="Claude">Claude</option> <option value="Grammarly">Grammarly</option> <option value="Gemini">Gemini</option> <option value="Elicit">Elicit</option> <option value="Writefull">Writefull</option> <option value="Llama">Llama</option> <option value="DeepL">DeepL</option> <option value="NotebookLM">NotebookLM</option> <option value="Grok">Grok</option> <option value="Qwen3">Qwen3</option> <option value="Mistral">Mistral</option> <option value="QwQ">QwQ</option> <option value="__other__" id="ai-option-other">Jiný nástroj</option> </select> </p> <p id="ai-tool-other-row"> <label id="ai-label-tool-other" for="ai-tool-other">Jiný nástroj, pokud není v seznamu</label><br> <input id="ai-tool-other" name="ai-tool-other" type="text" size="50" placeholder="Např. Midjourney"> </p> <table border="0" cellpadding="4" cellspacing="0"> <tr> <td><label id="ai-label-from" for="ai-period-from">Období použití od</label></td> <td><input id="ai-period-from" name="ai-period-from" type="date"></td> </tr> <tr> <td><label id="ai-label-to" for="ai-period-to">Období použití do</label></td> <td><input id="ai-period-to" name="ai-period-to" type="date"></td> </tr> </table> <p> <label id="ai-label-description" for="ai-description">Popište, jak byl nástroj použit</label><br> <textarea id="ai-description" name="ai-description" rows="6" cols="70" placeholder="Např. korektury textu, generování osnovy, pomoc s formulací rešeršních dotazů..."></textarea> </p> <p> <button id="ai-add-tool" type="button">Přidat nástroj do seznamu</button> </p> </fieldset> <br> <fieldset id="ai-recorded-section"> <legend id="ai-legend-recorded">Zaznamenaná použití</legend> <div id="ai-recorded-list"> <p>Zatím není zaznamenán žádný nástroj.</p> </div> </fieldset> <br> <fieldset> <legend id="ai-legend-data">Import/export</legend> <p id="ai-data-help">Stav formuláře můžete uložit do JSON souboru a později znovu načíst.</p> <p> <button id="ai-export" type="button">Exportovat JSON</button> <button id="ai-import" type="button">Importovat JSON</button> </p> </fieldset> <br> <fieldset> <legend id="ai-legend-result">Výsledné prohlášení</legend> <div id="ai-error"></div> <blockquote id="ai-result"> <p>Zde se po vyplnění formuláře zobrazí výsledné prohlášení.</p> </blockquote> <p> <button id="ai-copy" type="button">Kopírovat prohlášení</button> <button id="ai-clear" type="button">Vymazat formulář</button> </p> </fieldset> </div> <script> (function () { var entries = []; var lastText = ""; var resultIsPlaceholder = true; var AI_DECLARATION_LANGUAGE = 'cs'; function el(id) { return document.getElementById(id); } var T = { cs: { title: 'Prohlášení o využití AI', jsWarning: 'Tento formulář vyžaduje povolený JavaScript. Pokud po kliknutí na tlačítko nevznikne prohlášení, redakční systém pravděpodobně odstraňuje skripty. V takovém případě použijte tento formulář jen jako vodítko a výsledné prohlášení zapište ručně podle textů níže.', legendLang: 'Jazyk', labelLang: 'Jazyk prohlášení', langCs: 'Česky', langEn: 'English', legendUsed: '1. Použití AI', questionUsed: 'Použil/a jste při tvorbě této práce nějaký nástroj umělé inteligence?', usedNo: 'Ne, nepoužil/a jsem žádné nástroje AI.', usedYes: 'Ano, použil/a jsem alespoň jeden nástroj AI.', legendTool: '2. Použitý nástroj', labelTool: 'Nástroj AI', otherOption: 'Jiný nástroj', labelToolOther: 'Název jiného nástroje', otherPlaceholder: 'Např. Midjourney', labelFrom: 'Období použití od', labelTo: 'Období použití do', labelDescription: 'Popište, jak byl nástroj použit', descriptionPlaceholder: 'Např. korektury textu, generování osnovy, pomoc s formulací rešeršních dotazů...', addTool: 'Přidat nástroj do seznamu', legendData: 'Export/
```

Import', dataHelp: 'Rozpracovaný formulář si můžete uložit do JSON souboru a později znovu načíst.', exportJson: 'Uložit rozpracovaný formulář', importJson: 'Načíst rozpracovaný formulář', importOk: 'Data byla načtena.', importError: 'Soubor nemá platný formát nebo jej nelze načíst.', legendRecorded: 'Zaznamenaná použití', noEntries: 'Zatím není zaznamenán žádný nástroj.', remove: 'Odebrat', clear: 'Vymazat formulář', legendResult: 'Výsledné prohlášení', resultPlaceholder: 'Prohlášení se zde průběžně zobrazí podle aktuálního stavu formuláře.', copy: 'Kopírovat prohlášení', none: 'Prohlašuji, že při přípravě předložené práce nebyly použity žádné nástroje umělé inteligence.', usedIntro: 'Prohlašuji, že při přípravě předložené práce byly použity následující nástroje AI uvedenými způsoby:', usedOutro: 'Po použití uvedených nástrojů umělé inteligence jsem důkladně revidoval/a a upravil/a obsah podle potřeby a plně přejímám odpovědnost za výslednou podobu práce.', periodMissing: 'neuvedeno', descriptionLabel: 'popis použití', errorDescription: 'Vyplňte prosím popis použití nástroje.', errorTool: 'Uveďte prosím název nástroje.', errorPeriod: 'Datum „od“ nemůže být později než datum „do“.', ignoredEntries: 'Zaznamenaná použití jsou při volbě „Ne“ neaktivní a do prohlášení se nezahmou.', copied: 'Prohlášení bylo zkopírováno do schránky.', copyFailed: 'Kopírování do schránky se nezdařilo. Označte text ručně a zkopírujte jej. ', en: { title: 'AI Use Declaration', jsWarning: 'This form requires JavaScript. If no declaration appears after clicking the button, the content management system probably removes scripts. In that case, use this form only as guidance and write the final declaration manually using the text below.', legendLang: 'Language', labelLang: 'Declaration language', langCs: 'Česky', langEn: 'English', legendUsed: '1. AI use', questionUsed: 'Have you used any AI tool in writing this thesis?', usedNo: 'No, I did not use any AI tools.', usedYes: 'Yes, I used at least one AI tool.', legendTool: '2. AI tool used', labelTool: 'AI tool', otherOption: 'Other tool', labelToolOther: 'Name of the other tool', otherPlaceholder: 'e.g. Midjourney', labelFrom: 'Period of use from', labelTo: 'Period of use to', labelDescription: 'Describe how the tool was used', descriptionPlaceholder: 'e.g. text editing, outline generation, assistance with formulating literature-search queries...', addTool: 'Add tool to the list', legendData: 'Import/export', dataHelp: 'You can save the unfinished form to a JSON file and load it again later.', exportJson: 'Save unfinished form', importJson: 'Load unfinished form', importOk: 'Data has been loaded.', importError: 'The file is not valid or cannot be loaded.', legendRecorded: 'Recorded uses', noEntries: 'No tool has been recorded yet.', remove: 'Remove', clear: 'Clear form', legendResult: 'Generated declaration', resultPlaceholder: 'The declaration will be shown here automatically according to the current form state.', copy: 'Copy declaration', none: 'I declare that no AI tools were used in the preparation of this thesis.', usedIntro: 'I declare that the following AI tools have been used in preparation of this thesis in the ways indicated:', usedOutro: 'Having used the above AI tools, I have thoroughly revised and edited the content as necessary and take full responsibility for the final form of the thesis.', periodMissing: 'not specified', descriptionLabel: 'usage description', errorDescription: 'Please fill in the description of how the tool was used.', errorTool: 'Please enter the tool name.', errorPeriod: 'The “from” date cannot be later than the “to” date.', ignoredEntries: 'Recorded uses are inactive when “No” is selected and will not be included in the declaration.', copied: 'The declaration has been copied to the clipboard.', copyFailed: 'Copying failed. Please select the text manually and copy it.' }; function tr(key) { return T[getLang()][key]; } function getLang() { return AI\_DECLARATION\_LANGUAGE === 'en' ? 'en' : 'cs'; } function setText(id, value) { var node = el(id); if (node) node.textContent = value; } var otherToolRow = el('ai-tool-other-row'); var otherToolAnchor = otherToolRow ? otherToolRow.nextSibling : null; var otherToolParent = otherToolRow ? otherToolRow.parentNode : null; var otherToolInput = el('ai-tool-other'); function setOtherToolTexts() { var lang = getLang(); var label = otherToolRow ? otherToolRow.getElementsByTagName('label')[0] : null; if (label) label.textContent = T[lang].labelToolOther; if (otherToolInput) otherToolInput.placeholder = T[lang].otherPlaceholder; } function updateOtherToolVisibility() { var select = el('ai-tool'); if (!select || !otherToolRow || !otherToolParent) return; if (select.value === '\_\_other\_\_') { if (otherToolRow.parentNode) otherToolParent.insertBefore(otherToolRow, otherToolAnchor); if (otherToolInput) otherToolInput.disabled = false; setOtherToolTexts(); } else { if (otherToolInput) { otherToolInput.value = ""; otherToolInput.disabled = true; } if (otherToolRow.parentNode) otherToolRow.parentNode.removeChild(otherToolRow); } } function applyLanguage() { var lang = getLang(); setText('ai-title', T[lang].title); setText('ai-js-warning', T[lang].jsWarning); setText('ai-legend-lang', T[lang].legendLang); setText('ai-label-lang', T[lang].labelLang); var langSelect = el('ai-lang'); if (langSelect && langSelect.options.length >= 2) { langSelect.options[0].text = T[lang].langCs; langSelect.options[1].text = T[lang].langEn; } setText('ai-legend-used', T[lang].legendUsed); setText('ai-question-used', T[lang].questionUsed); setText('ai-label-used-no', T[lang].usedNo); setText('ai-label-used-yes', T[lang].usedYes); setText('ai-legend-tool', T[lang].legendTool); setText('ai-label-tool', T[lang].labelTool); setText('ai-option-other', T[lang].otherOption); setOtherToolTexts(); setText('ai-label-from', T[lang].labelFrom); setText('ai-label-to', T[lang].labelTo); setText('ai-label-description', T[lang].labelDescription); if (el('ai-description')) el('ai-description').placeholder = T[lang].descriptionPlaceholder; setText('ai-add-tool', T[lang].addTool); setText('ai-legend-data', T[lang].legendData); setText('ai-data-help', T[lang].dataHelp); setText('ai-export', T[lang].exportJson); setText('ai-import', T[lang].importJson); setText('ai-legend-recorded', T[lang].legendRecorded); setText('ai-clear', T[lang].clear); setText('ai-legend-result', T[lang].legendResult); setText('ai-copy', T[lang].copy); renderEntries(); if (resultIsPlaceholder) { el('ai-result').innerHTML = '<p> + htmlEscape(T[lang].resultPlaceholder) + '</p>'; } else { refreshDeclaration(); } updateOtherToolVisibility(); } function formatDate(value, lang) { if (!value) return ""; var parts = value.split('-'); if (parts.length !== 3) return value; if (lang === 'cs') return Number(parts[2]) + '.' + Number(parts[1]) + '.' + parts[0]; return Number(parts[2]) + '/' + Number(parts[1]) + '/' + parts[0]; } function formatPeriod(entry, lang) { var from = formatDate(entry.from, lang); var to = formatDate(entry.to, lang); if (from && to) return from + ' - ' + to; if (from) return lang === 'cs' ? 'od ' + from : 'from ' + from; if (to) return lang === 'cs' ? 'do ' + to : 'until ' + to; return T[lang].periodMissing; } function htmlEscape(s) { return String(s).replace(/&<>"/g, function (c) { return {'&': '&', '<': '<', '>': '>', '"': '"', '&#39;': '&#39;'}[c]; }); } function setError(message) { el('ai-error').innerHTML

```

= message ? '<p><strong>' + htmlEscape(message) + '</strong></p>' : ""; } function selectedUsed() { var radios
= document.getElementsByName('ai-used'); for (var i = 0; i < radios.length; i++) if (radios[i].checked) return
radios[i].value; return 'yes'; } function currentToolName() { var select = el('ai-tool'); if (select.value === '__other__')
return otherToolInput ? otherToolInput.value.replace(/^\s+|\s+$/g, "") : ""; return select.value; } function renderEntries()
{ var box = el('ai-recorded-list'); var lang = getLang(); var disabledByNo = selectedUsed() === 'no'; if (!entries.length)
{ box.innerHTML = '<p>' + htmlEscape(T[lang].noEntries) + '</p>'; return; } var html = disabledByNo ? '<p><em>'
+ htmlEscape(T[lang].ignoredEntries) + '</em></p>' : ""; html += '<ol>'; for (var i = 0; i < entries.length; i++) { html
+= '<li><strong>' + htmlEscape(entries[i].tool) + '</strong><br>' + htmlEscape(formatPeriod(entries[i], lang)) + '<br>'
+ htmlEscape(entries[i].description); if (!disabledByNo) { html += '<button type="button" data-ai-remove="' + i + '">'
+ htmlEscape(T[lang].remove) + '</button>'; } html += '</li>'; } html += '</ol>'; box.innerHTML = html; var buttons
= box.getElementsByTagName('button'); for (var j = 0; j < buttons.length; j++) { buttons[j].onclick = function ()
{ var index = Number(this.getAttribute('data-ai-remove')); if (!isNaN(index)) { entries.splice(index, 1); renderEntries();
refreshDeclaration(); } }; } function addEntry() { setError(""); var lang = getLang(); var tool = currentToolName(); var from
= el('ai-period-from').value; var to = el('ai-period-to').value; var description = el('ai-description').value.replace(/^\s+|\s+$/g, "");
if (!tool) { setError(T[lang].errorTool); return; } if (from && to && from > to) { setError(T[lang].errorPeriod); return; }
if (!description) { setError(T[lang].errorDescription); return; } entries.push({tool: tool, from: from, to: to, description:
description}); if (otherToolInput) otherToolInput.value = ""; updateOtherToolVisibility(); el('ai-period-from').value = ""; el('ai-
period-to').value = ""; el('ai-description').value = ""; renderEntries(); refreshDeclaration(); } function buildDeclaration()
{ setError(""); var lang = getLang(); if (selectedUsed() === 'no') { lastText = T[lang].none; resultIsPlaceholder = false;
return '<p>' + htmlEscape(lastText) + '</p>'; } if (!entries.length) { lastText = ""; resultIsPlaceholder = true; return '<p>' +
htmlEscape(T[lang].resultPlaceholder) + '</p>'; } var lines = [T[lang].usedIntro, ""]; for (var i = 0; i < entries.length; i++) {
if (lang === 'cs') lines.push('- ' + entries[i].tool + ' v období ' + formatPeriod(entries[i], lang) + ', ' + T[lang].descriptionLabel
+ ': ' + entries[i].description + '.'); else lines.push('- ' + entries[i].tool + ' in the period ' + formatPeriod(entries[i],
lang) + ', ' + T[lang].descriptionLabel + ': ' + entries[i].description + '.'); } lines.push(" T[lang].usedOutro); lastText
= lines.join('\n'); resultIsPlaceholder = false; var html = '<p>' + htmlEscape(T[lang].usedIntro) + '</p><ul>'; for (var
j = 0; j < entries.length; j++) { var item = lang === 'cs' ? '<strong>' + htmlEscape(entries[j].tool) + '</strong>' +
období <strong>' + htmlEscape(formatPeriod(entries[j], lang)) + '</strong>' + htmlEscape(T[lang].descriptionLabel)
+ ': ' + htmlEscape(entries[j].description) + '.' : '<strong>' + htmlEscape(entries[j].tool) + '</strong>' +
htmlEscape(formatPeriod(entries[j], lang)) + '</strong>' + htmlEscape(T[lang].descriptionLabel) + ': ' +
htmlEscape(entries[j].description) + '.'; html += '<li>' + item + '</li>'; } html += '</ul><p>' + htmlEscape(T[lang].usedOutro)
+ '</p>'; return html; } function refreshDeclaration() { var html = buildDeclaration(); if (html) el('ai-result').innerHTML =
html; } function clearAll() { entries = []; lastText = ""; resultIsPlaceholder = true; setError(""); el('ai-tool').selectedIndex
= 0; if (otherToolInput) otherToolInput.value = ""; updateOtherToolVisibility(); el('ai-period-from').value = ""; el('ai-period-
to').value = ""; el('ai-description').value = ""; el('ai-result').innerHTML = '<p>' + htmlEscape(tr('resultPlaceholder')) + '</p>';
renderEntries(); refreshDeclaration(); } function stateForExport() { var lang = getLang(); var used = selectedUsed(); var
oldEntries = []; for (var i = 0; i < entries.length; i++) { oldEntries.push({ toolId: entries[i].tool, customLabel: "", periodFrom:
entries[i].from || "", periodTo: entries[i].to || "", description: entries[i].description || "" }); } return { schema_version: 3,
generator: 'ai-declaration-cuni-embed', exported_at: new Date().toISOString(), lang: lang, used: used, done: true,
entries: oldEntries, embed_entries: entries.slice(), current: { toolId: "", customLabel: "", periodFrom: "", periodTo:
"", description: "", errors: {} }, answers: { q_usage: used === 'no' ? 'no' : 'yes', q_tools: oldEntries.map(function
(e) { return e.toolId; }), q_period_from: oldEntries.length === 1 ? oldEntries[0].periodFrom : "", q_period_to:
oldEntries.length === 1 ? oldEntries[0].periodTo : "", q_purpose: oldEntries.length === 1 ? oldEntries[0].description :
"" } }; } function downloadJson(filename, obj) { var text = JSON.stringify(obj, null, 2); var blob = new Blob([text],
{ type: 'application/json;charset=utf-8' }); var url = URL.createObjectURL(blob); var a = document.createElement('a');
a.href = url; a.download = filename; document.body.appendChild(a); a.click(); document.body.removeChild(a);
URL.revokeObjectURL(url); } function exportToJson() { downloadJson('ai-declaration-state.json', stateForExport()); }
function normalizeImportedEntry(entry) { if (!entry || typeof entry !== 'object') return null; var tool = entry.tool ||
entry.toolName || entry.name || entry.customLabel || entry.label || ""; if (!tool && entry.toolId) tool = entry.toolId;
if (entry.toolId === 'other' && entry.customLabel) tool = entry.customLabel; if (entry.toolId && entry.toolId !== 'other'
&& !entry.tool) tool = entry.toolId; tool = String(tool || "").replace(/^\s+|\s+$/g, ""); var description = entry.description ||
entry.usage || entry.purpose || ""; description = String(description || "").replace(/^\s+|\s+$/g, ""); var from = entry.from ||
entry.periodFrom || entry.period_from || entry.dateFrom || entry.q_period_from || ""; var to = entry.to || entry.periodTo
|| entry.period_to || entry.dateTo || entry.q_period_to || ""; if (!tool && !description) return null; return { tool: tool ||
T[getLang()].errorTool, from: String(from || ""), to: String(to || ""), description: description }; } function normalizeImport(obj)
{ if (!obj || typeof obj !== 'object') return null; var imported = { lang: obj.lang === 'en' ? 'en' : 'cs', used: 'yes', entries:
[] }; if (obj.used === 'no' || (obj.done === true && obj.entries && obj.entries.length === 0)) imported.used = 'no'; /*
Důležité: export nové verze kvůli zpětné kompatibilitě obsahuje současně embed_entries, entries i answers. Při importu
proto NESMÍME číst všechny větve najednou, jinak se nástroje zdvojí. Bereme první dostupný formát v pořadí: nový
embed formát -> původní entries -> staré answers. */ var source = null; if (Array.isArray(obj.embed_entries)) { source =
obj.embed_entries; } else if (Array.isArray(obj.entries)) { source = obj.entries; } if (source) { for (var i = 0; i < source.length;
i++) { var n = normalizeImportedEntry(source[i]); if (n) imported.entries.push(n); } } else if (obj.answers && typeof
obj.answers === 'object') { if (obj.answers.q_usage === 'no') imported.used = 'no'; var tools = obj.answers.q_tools;
if (!tools && obj.answers.q_tool) tools = [obj.answers.q_tool]; if (!tools && obj.answers.q_tools_other_text) tools

```

```

= ['other']; if (!Array.isArray(tools)) tools = []; var desc = obj.answers.q_purpose || obj.answers.description ||
"; var from = obj.answers.q_period_from || obj.answers.q_date_start || "; var to = obj.answers.q_period_to ||
obj.answers.q_date_end || "; for (var a = 0; a < tools.length; a++) { var toolId = tools[a]; var tool = toolId ===
'other' ? (obj.answers.q_tools_other_text || obj.answers.q_tools_other || "") : toolId; var e = normalizeImportedEntry({ tool:
tool, from: from, to: to, description: desc }); if (e) imported.entries.push(e); } if (imported.entries.length > 0 &&
imported.used !== 'no') imported.used = 'yes'; return imported; } function applyImportedState(data) { var normalized
= normalizeImport(data); if (!normalized) { setError(T[getLang()].importError); return; } el('ai-lang').value = getLang();
var radios = document.getElementsByName('ai-used'); for (var i = 0; i < radios.length; i++) radios[i].checked =
radios[i].value === normalized.used; entries = normalized.entries; el('ai-tool').selectedIndex = 0; if (otherToolInput)
otherToolInput.value = ""; el('ai-period-from').value = ""; el('ai-period-to').value = ""; el('ai-description').value
= ""; applyLanguage(); updateOtherToolVisibility(); toggleToolSection(); renderEntries(); refreshDeclaration();
setError(T[getLang()].importOk); } function importJson() { var fileInput = document.createElement('input'); fileInput.type
= 'file'; fileInput.accept = 'application/json,.json'; fileInput.setAttribute('hidden', 'hidden'); fileInput.onChange = function ()
{ var file = fileInput.files ? fileInput.files[0] : null; if (!file) return; var reader = new FileReader(); reader.onload = function
(event) { try { applyImportedState(JSON.parse(event.target.result)); } catch (err) { setError(T[getLang()].importError); }
if (fileInput.parentNode) fileInput.parentNode.removeChild(fileInput); }; reader.onerror = function ()
{ setError(T[getLang()].importError); if (fileInput.parentNode) fileInput.parentNode.removeChild(fileInput); };
reader.readAsText(file); }; document.body.appendChild(fileInput); fileInput.click(); } function copyResult() { var
lang = getLang(); if (!lastText) refreshDeclaration(); if (!lastText) return; if (navigator.clipboard &&
navigator.clipboard.writeText) { navigator.clipboard.writeText(lastText).then(function () { setError(T[lang].copied); },
function () { setError(T[lang].copyFailed); }); } else setError(T[lang].copyFailed); } function setSectionVisible(node,
visible) { if (!node) return; node.hidden = !visible; node.style.display = visible ? " : 'none'; } function toggleToolSection()
{ var showToolSections = selectedUsed() !== 'no'; setSectionVisible(el('ai-tool-section'), showToolSections);
setSectionVisible(el('ai-recorded-section'), showToolSections); renderEntries(); refreshDeclaration(); } if (el('ai-
lang')) el('ai-lang').value = getLang(); var warning = el('ai-js-warning'); if (warning && warning.parentNode)
warning.parentNode.removeChild(warning); el('ai-add-tool').onclick = addEntry; el('ai-export').onclick = exportJson;
el('ai-import').onclick = importJson; el('ai-clear').onclick = clearAll; el('ai-copy').onclick = copyResult; el('ai-
lang').onChange = applyLanguage; el('ai-tool').onChange = function () { updateOtherToolVisibility(); }; var radios
= document.getElementsByName('ai-used'); for (var i = 0; i < radios.length; i++) radios[i].onclick = function ()
{ toggleToolSection(); refreshDeclaration(); }; var validationIds = ['ai-tool-other', 'ai-period-from', 'ai-period-to', 'ai-
description']; for (var k = 0; k < validationIds.length; k++) { var validationNode = el(validationIds[k]); if (validationNode)
{ validationNode.oninput = function () { setError(""); }; validationNode.onChange = function () { setError(""); }; } }
applyLanguage(); updateOtherToolVisibility(); toggleToolSection(); refreshDeclaration(); }()); </script>

```

Jak generátor prohlášení funguje

Tento generátor prohlášení o užití umělé inteligence vám pomůže jednoduše shrnout, jaké nástroje umělé inteligence jste při tvorbě práce použili.

Nejprve odpovězte na otázku, jestli jste vůbec nějaké nástroje umělé inteligence použili. Pokud ne, je prohlášení velmi krátké a bez dalších kroků se vám vygeneruje ve spodní části stránky, v sekci **Výsledné prohlášení**.

Pokud jste nějaký nástroj AI použili, vyplňte, o jaký se jedná, v jakém časovém období jste jej využívali a jakým způsobem. Kliknutím na **Přidat nástroj do seznamu** se Vám tento záznam přidá do Výsledného prohlášení ve spodní části stránky a zároveň vymaže zadané informace k tomuto nástroji.

Následně můžete zadat jiný nástroj stejným způsobem, který se vám opět kliknutím na **Přidat nástroj do seznamu** přidá do **Výsledného prohlášení** ve spodní části stránky.

V sekci **Zaznamenaná použití** si můžete průběžně prohlížet a kontrolovat seznam vámi zadaných nástrojů a můžete zde nástroje odebírat.

Pozor, kliknutím na **Vymazat formulář** ve spodní části stránky se vymažou všechna zaznamenaná použití.

Rozpracovaný formulář si můžete i uložit a vrátit se k němu později.

Uděláte to jednoduše. V sekci Import/export vyberte tlačítko Exportovat JSON. Tento dokument si stáhněte do počítače. Až se budete chtít k práci na formuláři vrátit, opět otevřete tuto stránku a tentokrát v sekci Import/export vyberte Importovat JSON a do stránky vložte uložený dokument.

Rozpracovaný formulář se vám opět načte.

Až budete s podobou Prohlášení spokojeni a budete si jisti, že jste na žádný použitý nástroj nezapomněli, zkopírujte výsledné prohlášení a vložte jej na příslušné místo ve vaší závěrečné nebo jiné práci.

Příklady Prohlášení o užití umělé inteligence

Prohlašuji, že při přípravě předložené práce byly použity následující nástroje AI uvedenými způsoby:

- Grammarly v období 12. 4. 2026 – 12. 5. 2026, popis použití: Jazyková a stylistická revize rukopisu, zejména oprava gramatických chyb a zlepšení čitelnosti textu..
- ChatGPT v období 8. 3. 2026 – 30. 5. 2026, popis použití: Vytvoření počáteční struktury kapitol, brainstorming nad interpretačními přístupy k analyzovanému textu a k provedení základní rešerše dostupných zdrojů..

Po použití uvedených nástrojů umělé inteligence jsem důkladně revidoval a upravil obsah podle potřeby a plně přejímám odpovědnost za výslednou podobu práce.

Prohlašuji, že při přípravě předložené práce byly použity následující nástroje AI uvedenými způsoby:

- GitHub Copilot v období 20. 2. 2026 – 27. 3. 2026, popis použití: Generování a optimalizace programového kódu, vytváření kostry algoritmu a návrhu testovacích funkcí..
- Claude v období 7. 2. 2026 – 2. 5. 2026, popis použití: Vysvětlení komplexních konceptů strojového učení, diskuse nad architekturou neuronových sítí a revize technických vysvětlení v textu..
- Microsoft Copilot v období 27. 2. 2026 – 23. 5. 2026, popis použití: Vytvoření grafů na stránkách 28, 39 a 41 a k vygenerování rejstříku..

Po použití uvedených nástrojů umělé inteligence jsem důkladně revidoval/a a upravil/a obsah podle potřeby a plně přejímám odpovědnost za výslednou podobu práce.

Prohlašuji, že při přípravě předložené práce byly použity následující nástroje AI uvedenými způsoby:

- ChatGPT v období 17. 5. 2026 – 30. 5. 2026, popis použití: Vytvoření osnovy práce, syntéza současného stavu vědeckých poznatků a výběr relevantní odborné literatury..
- Microsoft Copilot v období 28. 2. 2026 – 21. 3. 2026, popis použití: Konzultace výběru analytického postupu, vytváření grafů a tabulek jazyková formulace vlastní interpretace výsledků..
- Grammarly v období 6. 3. 2026 – 10. 5. 2026, popis použití: Jazyková korektura a formátování textu.

Po použití uvedených nástrojů umělé inteligence jsem důkladně revidovala a upravila obsah podle potřeby a plně přejímám odpovědnost za výslednou podobu práce.

Prohlašuji, že při přípravě předložené práce byly použity následující nástroje AI uvedenými způsoby:

- ChatGPT v období 10. 1. 2026 – 13. 2. 2026, popis použití: Brainstorming nad výzkumnými otázkami, strukturování teoretických konceptů a diskuse nad různými sociologickými perspektivami..
- Claude v období 10. 7. 2026 – 12. 8. 2026, popis použití: Analýza kvalitativních dat z rozhovorů a vytváření tematických kategorií..
- Notion AI v období 10. 7. 2026 – 20. 8. 2026, popis použití: Organizace poznámek a vytvoření konceptuální mapy..

Po použití uvedených nástrojů umělé inteligence jsem důkladně revidoval/a a upravil/a obsah podle potřeby a plně přejímám odpovědnost za výslednou podobu práce.