

# **Website renovation**

**Technical specification for development and CMS – Lot 2**

Date: 13. 5. 2026

Version: final tender document

## Table of Contents

1	Website renovation and software leasing .....	5
1.1	Context and objectives of the renovation .....	5
1.2	Purpose and scope of the project .....	6
2	Important technological pillars of the CMS system .....	8
3	Basic criteria for choosing a CMS .....	9
3.1	Modularity - CMS management and reward .....	10
3.1.1	Adding/Changing Web Pages .....	10
3.1.2	Adding and Managing Content Modules/Document Types .....	11
3.1.3	Add and change content blocks web page widgets .....	12
3.1.4	Adding/changing templates.....	12
3.1.5	Add or change web forms .....	13
3.2	Centralized management of multiple sites.....	13
3.2.1	Domains and languages at the site level .....	13
3.2.2	Centralized access and privilege management .....	14
3.3	Multilingualism .....	16
3.4	Other functional requirements.....	16
3.4.1	Versions/change history.....	16
3.4.2	Editorial dashboard .....	17
3.4.3	User management.....	17
3.4.4	Preview and publish with a delay .....	18
3.4.5	Publishing and managing media content.....	18
3.4.6	QR Code Generation.....	18
3.5	Connectivity .....	19
3.6	Use of artificial intelligence and personalization .....	19
4	Hosting & Security.....	19
4.1	Safety and compliance of the development environment .....	20
4.2	CMS security mechanisms .....	21
5	Tasks of Lot 2 – Website development.....	21
5.1	Establishment of a development environment (DEV, S TAGE, PROD) .....	21
5.2	Implementation of CMS foundations (structure, content types / modules) .	22

5.3	Implementation of visual templates (frontend development) .....	22
5.3.1	Odzivnost (Responsive implementation) .....	23
5.3.2	Accessibility .....	24
5.3.3	SEO, LLM in Core Web Vitals optimizacija .....	25
5.3.4	Analytics .....	26
5.4	Backend implementation and integrations .....	26
5.4.1	Implementation of integrations .....	27
5.4.2	Compliance .....	28
5.4.3	Cookie banner .....	29
5.5	Content migration – automated migration (contractor) and manual migration (UX agency) .....	29
5.5.1	Preparation of automated migration flows according to the Migration Playbook .....	29
5.5.2	Implementation of Redirects (301 Redirect Map) .....	30
5.5.3	Prepare your environment and tools for manual migration .....	30
5.5.4	Final validation of the total migration volume .....	30
6	Testing (QA, WCAG, SEO, performance and security checks) with demarcation of competence .....	32
7	Introduction to production (Go-Live) .....	33
7.1	Tasks of the DEVELOPMENT TEAM .....	33
7.1.1	Preparation and approval of the Go-Live plan (Runbook) .....	33
7.1.2	Go-Live deploy .....	34
7.1.3	Technical support at the time of launch .....	34
7.2	Tasks of the client .....	34
7.3	Post-Go-Live Stabilization (Post-Go-Live) .....	34
8	Final documentation, training and handover of the system .....	34
8.1	Preparation of complete technical documentation .....	35
8.2	Preparation of editorial documentation .....	35
8.3	Training for editors (DEVELOPMENT AND UX TEAM) .....	35
8.4	Technical training for the client's IT team (DEVELOPMENT TEAM) .....	36
8.5	Final validation and signature record .....	36
9	Download the code .....	36
10	UX and Development Team Collaboration .....	36
11	Roles and responsibilities of the project team .....	37

12	Key project milestones and payment dynamics .....	39
12.1	Explanations .....	40
12.1.1	Definition of a completed milestone .....	40
12.1.2	Milestone review and validation process .....	40
12.1.3	Criteria for confirming a milestone.....	41
12.1.4	Milestone-to-payment ratio .....	41
12.1.5	The role of the independent expert .....	41
12.1.6	Moving to the next milestone .....	41
<b>13</b>	<b>Maintenance</b> .....	46
13.1	Warranty during the warranty period .....	46
13.2	Maintenance .....	46
13.2.1	Routine technical maintenance, covering in particular: .....	47
13.2.2	Regular Support, which includes, in particular: .....	47
13.2.3	Upgrades (Change Requests), which includes, for example: .....	49

# 1 Website renovation and software leasing

SPIRIT Slovenia is a public agency of the Republic of Slovenia and the central point of contact for companies and investors. Its mission is to promote the growth, competitiveness and international visibility of the Slovenian economy through coordinated support programs, consulting, promotion and integration of key development actors. SPIRIT connects entrepreneurship, internationalization, attracting foreign investment and technological progress, and represents Slovenia on global markets as a country that is green, creative and smart – *Green. Creative. Smart.*

The purpose of the SPIRIT Slovenia website renovation and consolidation project is to establish a single, multilingual and accessible to all (WCAG 2.2 AA) digital platform, which will be simple to manage at its core, but powerful enough to effectively connect companies, investors and ecosystem partners. The new solution will provide users with up-to-date, consistent and easy-to-find information, and editorial teams with a transparent, automated and data-driven environment for their daily work.

The renovation represents an important step towards the digital transformation of public services and is a direct realization of **the promise of "Green. Creative. Smart."**

- **Green:** By merging four existing websites into one multi-party platform, we want to reduce energy consumption, data transfer and server load. We are introducing EU hosting on renewable resources, lighter sites ( $\leq 700$  KB), smart media management (AVIF/WebP).
- **Creative:** a unified design system will enable a consistent, visually recognizable and content-rich presentation of the Slovenian economy and contribute to the comprehensive digital identity of SPIRIT Slovenia.
- **Smart:** standardized editorial processes, clear assignment of roles and rights, automation of translations, inventories and redirection will ensure higher efficiency, transparency and quality control.

The result of the project will be **a sustainable, inclusive and data-driven digital experience** that will support the development and internationalization of the Slovenian economy in the long term, while strengthening the role of SPIRIT Slovenia as a central institution for supporting companies and investors.

## 1.1 Context and objectives of the renovation

SPIRIT Slovenia currently operates 4 independent websites on four different platforms, with different integrations and data sources. Each of them has its own history, structure, visual identity and tone of communication. As a result, there is brand dispersion, duplication of content, different user experiences, and inefficient management.

The renovation and consolidation project goes beyond mere technical unification. It is a **content and visual merger of hitherto independent websites, and thus**

**individual brands, under the umbrella brand SPIRIT Slovenia.** The new platform will represent a single digital environment where all content and services of SPIRIT will be communicated consistently, transparently and in the spirit of its identity "Green. Creative. Smart."

The technological goal of the project is to consolidate all four websites into one modern CMS platform with multilingual and *multisite* architecture.

With the renovation and consolidation of websites, we want to achieve:

- content and visual unification and clear transfer of the values and narrative of the umbrella brand SPIRIT,
- maintaining the functional and substantive independence of individual portals, where it makes sense from the point of view of the target groups,
- a uniform, stable and energy-efficient infrastructure that will unify operations in the long term, reduce maintenance costs and provide editors with a modern and transparent working environment,
- management of all production and development versions of the system within its own infrastructure, with a high level of security, transparency and scalability.
- standardized editorial workflows (roles, rights, revisions, archiving, translations),
- optimized accessibility, usability and compliance with WCAG 2.2 AA and GDPR standards,
- A next-generation SEO strategy, also adapted to information retrieval through artificial intelligence (LLM),
- Measurable goals, built-in analytics, and reliable content migration with less than 1% of dropped links.

The new solution will be the foundation **of SPIRIT Slovenia's unified digital presence**, which will offer users a unified story and experience, regardless of which area or service they are looking for.

## 1.2 Purpose and scope of the project

The project is divided into two parts:

**Lot 1: UX & Design** – strategic consolidation design, catalogue of key functionalities (FE/BE), design system and prototypes, content inventory with Redirect Map and Migration Playbook, editorial workflows and standards (WCAG 2.2 AA, SEO, GDPR).

**Lot 2: Development** – implementation of solutions in the CMS tool (must meet the standards defined in the technical specification), integration, migration and automation, establishment of environments and control, security and implementation of testing and deployment.

The SPIRIT Slovenia website consolidation and replatforming project represents a key step towards establishing a modern, centralized and secure digital environment that will enable the efficient management of all websites under the SPIRIT umbrella brand. The purpose of the technological renovation is to provide a single, stable and energy-efficient platform that will unify operations in the long term, reduce maintenance costs and provide editors with a modern and transparent working environment.

With the consolidation and transition to a single CMS system, the client wants to:

- **Improve the user experience of editing** – for both internal and external editors, who will edit content within a single, logically structured and user-friendly platform,
- **establish full control and ownership of all elements of the system**, including code and data, and ensure transparent and controlled content management,
- **centralize the hosting of all websites on its own infrastructure**, which will allow for lower costs, greater security and energy efficiency of the CMS system.

The technological goal of the website renovation is the technological unification of all websites within a single CMS platform, installed and maintained in the client's environment (initially "on-premises", later presumably "cloud"). Sites that are combined<sup>1</sup>:

**spiritslovenia.si** – corporate website of the public agency SPIRIT (information about the agency, programs and contacts), intended to present the mission and direction of the agency. Working estimate ranges: ~200-400 URLs.

**izvoznookno.si** – a portal for exporters with content on foreign markets, opportunities, news and events ("Web of Information and Opportunities for Exporters"). Working estimate ranges: ~4,000–8,000 URLs.

**podjetniski-portal.si** – an information centre for potential and existing entrepreneurs (calls, events, news, programmes, manuals), with rich archives of publications. Working estimate ranges: ~6,000-10,000 URLs.

**sloveniabusiness.eu** – the central B2B portal for investors and buyers from abroad ("Invest in Slovenia", "Buy from Slovenia"), with news, events, thematic sections (industries, business environment, success stories) and multilingual navigation. Working estimate ranges: ~1,200-2,000 URLs.

---

<sup>1</sup> More detailed information on each website can be found in **Annex 1: Existing websites**.

## 2 Important technological pillars of the CMS system

For the successful consolidation and long-term management of all SPIRIT Slovenia websites, it is crucial that the selected CMS is based on a modern, scalable and secure architecture. The system must allow for uniform administration, multilingualism, a high level of security and easy integration with other information systems.

The basic orientations are summarized in five technological pillars, which represent the framework for the selection and design of the CMS solution. More detailed technical requirements are defined in the following chapters.

### I. MODULARITY

The CMS should be based on a modular architecture that allows easy addition, modification and upgrade of individual functionalities without the need for custom development. The system must allow for rapid adaptation to changes and efficient management of substantive and functional sets.

### II. CENTRALIZATION

The new solution should enable the management of all four websites within a single CMS instance (*multi-site* model). A centralized approach will allow editors to work easily, reduce duplication of content, and make it easier to control quality and updates.

### III. MULTILINGUALISM

The CMS must support multilingual editing and localization of content within a single platform. Editors need to have an overview of translations, content statuses, and the ability to use common elements across languages and portals.

### IV. SAFETY

The system must be designed according to the principle of "*secure by default*" – with built-in protection mechanisms at the level of the CMS, database and infrastructure. User access, data and content must be protected in accordance with recognized good practices in this field and with the law governing the field of information security in the Republic of Slovenia (the client is a provider of essential services).

### V. CONNECTIVITY

The CMS must enable reliable and secure data exchange with other information systems of SPIRIT Slovenia and external platforms through standardized API interfaces (REST, GraphQL...). Connectivity is key to building an effective digital ecosystem.



### 3 Basic criteria for choosing a CMS

A suitable CMS system represents a fundamental technical and strategic step in the establishment of a new digital platform SPIRIT Slovenia. The chosen solution must enable long-term stability, security, flexibility and effective editorial work, while at the same time being in line with the principles of energy efficiency, sustainability and control over your own data.

The CMS must be **a proven, modern and established solution**, suitable for complex multilingual and multi-portal environments, such as those managed by SPIRIT Slovenia. The proposed CMS must have long-term technical support and development continuity (LTS) and a wide partner network of contractors that allows the independence of the client in the selection of the contractor.

#### Criteria for choosing a CMS

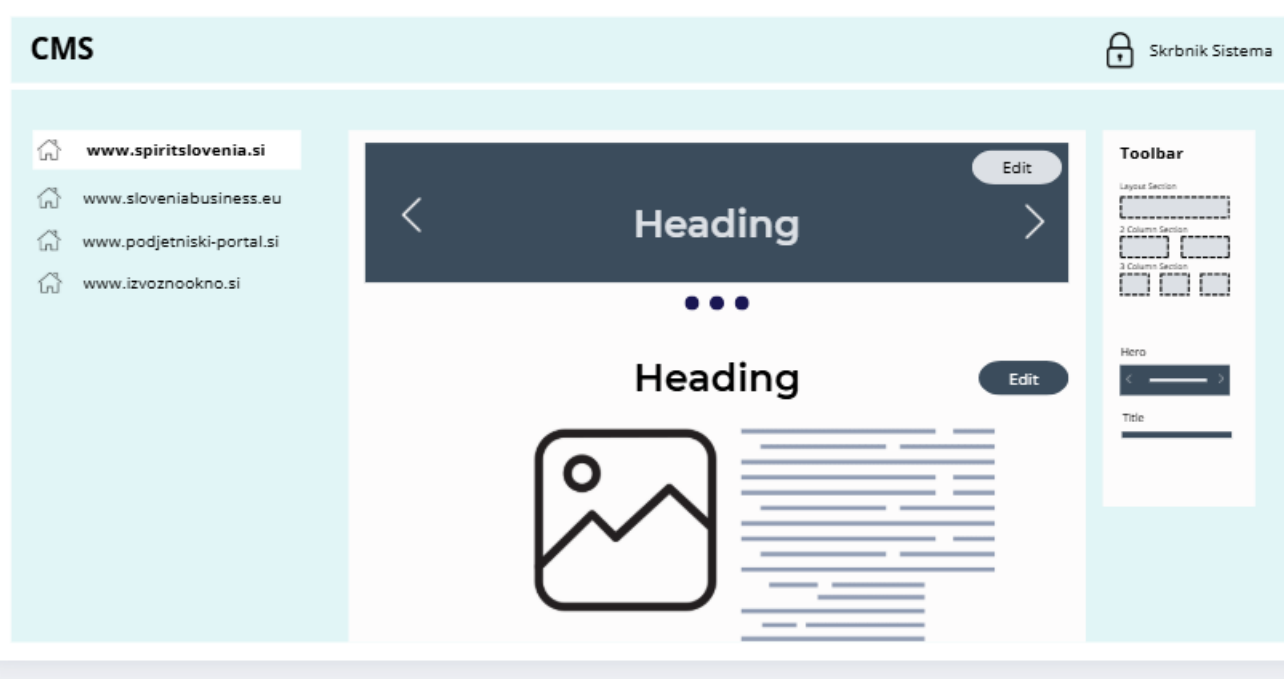
- 1) **Flexibility and scalability:** The CMS must enable modular expansion of functionality without the need for custom development and without compromising the stability of the system. The structure should allow for the easy introduction of new content types, templates, and components.
- 2) **Default functionalities ("out-of-the-box"):** The system must include a wide range of standard functionalities (content editing, versions, translations, SEO, rights, revisions, forms, etc.) without the use of external plug-ins.
- 3) **User interface (UI/UX for editors):** The CMS must have a modern, transparent and intuitive editorial interface that allows for easy content editing, multi-portal work and fast workflows.
- 4) **Maturity and adoption of the solution:** The CMS must be among the proven and established solutions on the European market (e.g. present in the *Mid-Market Europe for Web Content Management Software* category according to [the https://www.g2.com](https://www.g2.com) portal).
- 5) **Speed and efficiency:** The solution must meet high standards of performance and optimization, and be able to load quickly and responsively, even for large-scale multilingual content.
- 6) **Security and reliability:** The CMS must be designed according to *the principle of secure by default*, with regular LTS updates, built-in protection mechanisms (OWASP Top 10) and audit traceability.
- 7) **Long-term support (LTS):** The system must provide long-term version-specific support (LTS) and a clearly defined update and migration plan.
- 8) **Documentation and openness:** Publicly available, official and complete technical documentation for all current versions of the CMS (API, data structure, integration procedures) must be available.
- 9) **Ownership and Independence:** The CMS must be able to own code and data and operate on the client's infrastructure (*on-premises*). The architecture must ensure the portability of the solution and the possibility of switching contractors without loss of data or functionality.

- 10) **Support and partner network:** The CMS must have an active developer community and a broad partner network with official manufacturer support, which provides the client with continuity of support, the possibility of choosing different contractors and independence from each provider.

## 3.1 Modularity - CMS management and reward

### 3.1.1 Adding/Changing Web Pages

The CMS system should allow website editors to arbitrarily build new websites or rework existing pages, with the help of pre-prepared content sets/building blocks that will be prescribed by the UX rules.



*The figure shows a general demonstration of operation and does not represent a requirement for the appearance of the CMS interface*

To manage the content of each website, a so-called wysiwyg ("What you see is what you get") website editor may be available, which must be available **without the installation of additional "third party plugins"**

Functional modifications of the offered CMS system **are allowed**, which will make it easier for editors to build pages with the help of UX prescribed and pre-prepared building blocks.

The website editor should have the following functionalities:

- Drag-Drop/Drag & Drop – Add/change the position of content sections (Section, Container)

- Drag-Drop/Drag & Drop - Adding sections with multiple columns in the Grid - taking into account the construction according to [the https://www.w3.org/TR/css-grid-1/ standard](https://www.w3.org/TR/css-grid-1/)
- Adding/changing the content of content blocks that must be prepared on the basis of approved UX regulations
- Translation of individual assemblies / building blocks
- Turn off/side-by-side comparisons of two languages
- Change the order, sections, and building blocks
- Delete sections and Widgets

### 3.1.2 Adding and Managing Content Modules/Document Types

The CMS must **allow content modules such as: (News, Events, Programs...) to be added and changed within the CMS interface for managing modules/document types** without installing an additional plugin **or** additional processing of the CMS system.

The CMS Module Management Interface should enable the following functionalities:

- Adding a new module/document type
- Changing and adding new input fields/attributes in existing modules
- Changing the type of an existing input field (e.g.: from Text to LongText, ...)
- Add and manage input fields of the following types
  - Text/String,
  - LongText,
  - Editor/Editor,
  - Image,
  - Slike/Image Gallery,
  - Dropdown list,
  - CheckBox List
  - Number,
  - Web address
  - List of URLs
  - Datum in Ura/date & hour,
  - Da|Ne/Yes|No,
  - Upload/Media Upload,
  - Tags,
  - Blok koda/Code Block,
  - Relational link/Related type... (1:1 or 1:1 link)
- Automatic construction of attributes in the data layer of the application after publishing a change on the selected Module
- Adding new fields to a Template or Individual Page either through the Visual Code Editor within the CMS or through code processing using an external coding tool

### 3.1.3 Add and change content blocks web page widgets

The CMS system should provide a simple and intuitive way of editing in accordance with UX rules. All building block containers must follow the UX rules of content display. The CMS should allow the site administrator to easily rework or build container widgets using a centralized management interface.

The CMS interface for managing content parts must enable the following functionalities:

- Reworking and creating a new content block
- Add changes to the attributes of each content part
  - o Text/String,
  - o LongText,
  - o Editor/Editor,
  - o Image,
  - o Slike/Image Gallery,
  - o Dropdown list,
  - o CheckBox List
  - o Web address
  - o List of URLs
  - o Number,
  - o Datum in Ura/date & hour,
  - o Da|Ne/Yes|No,
  - o Upload/Media Upload,
  - o Tags,
  - o Blok koda/Code Block,
  - o Relational content link/Related type... (1:1 or 1:1 link)
- Add a content part to the webpage editor toolbar
- Option to visually rework a widget to display on the toolbar
- Automatically build attributes in your app's data layer when a change is published on the selected widget

### 3.1.4 Adding/changing templates

The CMS System should enable easy separation of formatted regulations at the level of the website or individual website through Templates or Themes.

Each website or web page must belong to a design template/Theme within a single CMS environment, in which the prescribed format is specific to a particular website or web page

The CMS system should allow website editors to:

- Arbitrarily build new templates using the template management interface or using the coding tool

- Easily change the link between the template and an individual web page – every web page should be based on a template
- easy change of templates of individual modules / document types

### 3.1.5 Add or change web forms

The CMS must allow website editors to build and modify Web forms for communication with website users (Contact Form, Newsletter Form, Event Registration...).

The CMS web form editor should have the following functionalities:

- Add and change the headings and descriptions of individual sections of a form
- Add modifying data fields of the following types
  - o Kratek Text
  - o Dolg Text
  - o Date
  - o Option selection
  - o Download a file
  - o Password
  - o Privolitev/Consent
  - o Drop-down list
  - o Text editor
  - o reCAPTCHA
- Send a notification about a new record to the address of the selected editor
- The ability to add and link to any thank-you page
- Review submitted form data
- Sort and filter sent data
- Export sent data to Excel/CSV
- Option to copy the entire form
- Possibility of integration with back-end systems – WebHook or similar
- Enabling versioning at the level of individual pages, articles and other content sets
- Easily add a translation of the existing selected content of a page, article, event and other content sets.

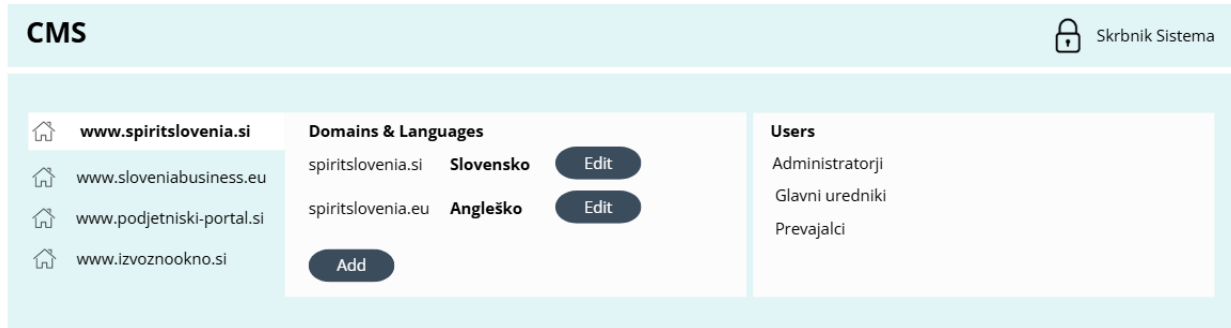
## 3.2 Centralized management of multiple sites

### 3.2.1 Domains and languages at the site level

The system should allow you to add and change links between an individual website and a domain, without the intervention of an agency or changing the code in the backend of the system.

Changes to the links between the domain and the website can only be managed by the CMS administrator in the interface dedicated to managing domains and connecting to content.

Example interface:



*The figure is a general representation of the operation and does not represent a requirement for the appearance of the CMS interface*

If the DNS server side has already arranged the entry of the appropriate DNS records to the new domain, the published changes to the new domain must be displayed in the CMS interface immediately after the content related to the new domain is published.

The system must be able to easily and centrally manage the connections of the language to the domain or directory: e.g.:

- domena.de = DE language (link domain -> language>),
- domena.si/de = DE language (directory-> language link>)

See: INTERFACE EXAMPLE – Domains & Languages

The system must be able to turn on the display of an alternative language in the event that a primary translation is not available.

The system must allow for easy and advanced search (search, filtering, sorting) between all languages and content sets.

### 3.2.2 Centralized access and privilege management

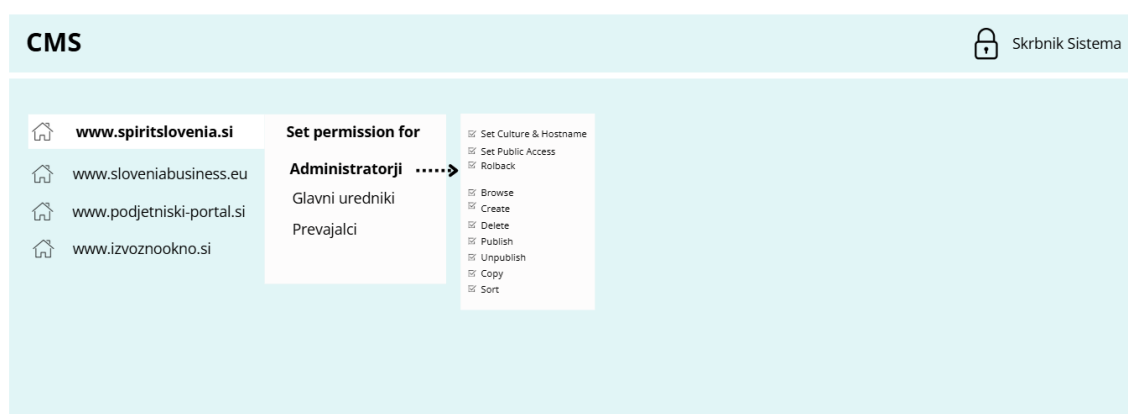
By default, the CMS should allow:

- Centralized and easy management of users and user groups.
- Separation of users at the level of access to the editorial interface and content.
  - Editors/Administrators – allowed access and editing
  - Users – Allowed access to published protected content
- The use of external providers for online logins (OAuth 2.0, Open ID, SAML 2.0) such as Google's OpenId, Microsoft identity platform and others... and the ability to integrate with Entra ID/Azure Active Directory.

By default, the user and rights management user interface must allow:

- Adding/deleting individual editors
- Change passwords for individual editors
- Adding/Removing User Groups
- Adding/removing a user without access to the editorial interface and the ability to specify access to certain protected content published on the site (see Figure: Determining the rights of an individual page)
- Adding/removing rights at different content levels:
  - o Level - A web page or tree structure of a page (see example: Defining the rights of an individual page)
  - o Level - A building block of an individual web page (e.g.: News Section, Home page)
  - o Level - Attribute of each widget (e.g.: Title of the News section)

For example, to set the rights of an individual page:



*The figure shows a general demonstration of operation and does not represent a requirement for the appearance of the CMS interface*

## 3.3 Multilingualism

By default, the system should allow site editors to easily add content for the selected language within the CMS system. Translations of individual content must be accessible in one place and allow direct mapping of the content.

Example: Content mapping for language versions:

The screenshot shows a CMS interface with a light blue header. On the left, there is a sidebar with a home icon and the URL 'www.spiritslovenia.si'. The main content area is divided into two columns. The left column is titled 'Slovenian' and contains a 'Title' field with the text 'Slovensko besedilo'. The right column is titled 'English' and contains a 'Title' field with the text 'English text'. In the top right corner of the header, there is a lock icon and the text 'Skrbnik Sistema'.

*The figure shows a general demonstration of operation and does not represent a requirement for the appearance of the CMS interface*

Input of translations for all added languages must be enabled by default at all content levels of the system: (Page title, Page content, SEO descriptions, Navigation, Buttons, Content widgets...).

By default, the CMS system should allow the input of translations of system labels/tags that are used in the code to display the contents.

The CMS should be able to:

- marking the status of the translation ("missing", "in translation", "translated"),
- warnings for unsynchronized translations after the original has been modified,
- automatic inheritance of metadata and SEO fields between language versions.

The translator must be able to work in **split view**, where he sees the original and the translation at the same time.

## 3.4 Other functional requirements

### 3.4.1 Versions/change history

- By default, the CMS should provide insight into the audit trail of changes at the level of individual pages.
- By default, the CMS should allow the editor to revert past published content.



- To better represent the changes, the system must be able to display a comparison between the currently published content and the previously selected content.
- For details, see Lot 1 - Editorial workflow plan
- Part of the editorial flows must be tied to **the mandatory tagging of content with a validated taxonomy (see Lot 1 step C)**.
- The CMS must disable the publication of content without at least one mandatory tag (e.g. topic, target group, content type).
- Adding new tags is only possible with the approval of the administrator.
- Editorial flows must include automated checks for:
  - manjkajoče SEO elemente (title, meta description),
  - missing ALT image descriptions,
  - contrast discrepancies (WCAG),
  - missing translations,
  - missing taxonomic tags.

### 3.4.2 Editorial dashboard

The editorial dashboard must enable at least:

- overview of content by status (draft, under review, translated, published, archived),
- Number of content by language
- review of unpublished and expired content,
- Warnings about missing translations, taxonomic tags, and other errors
- and expired posts.

### 3.4.3 User management

For the purposes of user management, the CMS must enable by default:

- Overview of registered users within the CMS solution
- Searching for a user
- Editing/changing individual user's data
- Change an individual user's password
- Delete an individual user
- Assign a user to a Presented Rights group
- Insight into the security log (Successful/Unsuccessful logins to the system)

#### 3.4.4 Preview and publish with a delay

- The CMS must allow the site editor to preview changes before publication and to temporarily store content before public release.
- The CMS must allow the site editor to publish content with a delay. E.g.: Post on Day: Hour.
- The CMS must allow the site editor to remove published content with a delay. Eg: Remove Post On Day: Clock

#### 3.4.5 Publishing and managing media content

- The CMS system must allow the website editor to download and store at least the following content:
  - Slika (.jpg, jpeg, webp, .png, .svg)
  - File (.pdf, doc, docx, .xlsx, .xls, .csv)
  - Audio files (.mp3, .web, .oga, .opus)
  - Video file (.mp4, .webm, .ogv)
- The CMS interface should allow the editor to freely build and manage the tree structure of folders within which the media content will be stored.
- The CMS interface must enable the following functions for managing individual media content:
  - replacement of existing media content
  - Entering/Changing Alt Tags
  - entry/change Title Tags
  - Deletion
  - Copying
  - SexySexy
- The CMS system should allow the site administrator to easily add rules for automatic cropping and sizing of images. Cropping and resizing rules must be followed when uploading images in widgets on which an editorial rule is assigned. It is desirable that the system allows you to add a watermark to the image to protect copyright.
- The CMS must allow the Site Administrator to add any new type of media content.

#### 3.4.6 QR Code Generation

- The CMS system should allow for refinement, which allows the creation of QR codes for all published content.
- The QR code generator should allow easy settings of the dimensions and format of the QR code creation.

- For each QR code generated, there must be a downloadable option in the format (.jpg, png)

### 3.5 Connectivity

The CMS should be able to easily connect to external data sources and applications ("API-friendly by design") using API interfaces and Webhooks

By default, at least the following data exchange protocols (REST, SOAP, and GraphQL) must be supported.

By default, the CMS system should allow you to instantly respond to changes in content by sending requests when the selected "Webhooks" event occurs.

By default, the CMS must be able to store content in all existing and new CMS data entities via API interfaces.

To protect the connection, the CMS must provide the following security mechanisms:

- Locking/unlocking API access to selected CMS data sources
- Protect access with Authorization
- Limit the number of calls
- Možnost nadzora pravil: Cross-Origin Resource Sharing (CORS)

### 3.6 Use of artificial intelligence and personalization

To use AI tools and integration, a CMS must be able to.

- Easy storage and operation of code from AI cloud solutions
- Easy installation of AI extensions of the CMS system (Compilers, process optimization...)
- Possibility of using the MCP system  
<https://modelcontextprotocol.io/docs/getting-started/intro> as an independent part of the CMS solution
- Possibility of easy implementation of the tool for Website Personalization with the help of AI

## 4 Hosting & Security

For the needs of hosting both development versions and production versions of the CMS solution, **the client will provide the CMS solution** provider with the following infrastructure:

- Windows Server (2025)
- IIS 10.0 + .NET 8 or later
- SQL Server 2022 or later
- Cloudflare DNS

The contractor can customize the CMS solution on its own infrastructure, but it must ensure easy transfer of the code to the above-mentioned server environment **without installing additional server components.**

The contractor must follow the guidelines for ensuring comprehensive cybersecurity, including the guidelines defined in the OWASP Top Ten, already at the planning stage.

## 4.1 Safety and compliance of the development environment

The programming language used with the help of which the offered CMS is developed and the language with the help of which the customization of the CMS system is carried out must:

- Be defined as an MIT OpenSource programming language listed in ([https://en.wikipedia.org/wiki/List\\_of\\_open-source\\_programming\\_languages](https://en.wikipedia.org/wiki/List_of_open-source_programming_languages)) – for the purpose of easy transfer of ownership
- Be a strongly typed and object-oriented programming language (Strongly typed/Object oriented) to ensure the highest level of security
- Enable Code Compiling - for the purpose of optimizing stable and fast operation
- Enable hosting on the existing SPIRIT server infrastructure – for cost optimization purposes
- Have a clearly defined update process (LTS) accessible on the CMS provider's publicly accessible website - for the purpose of achieving the prescribed level of compliance
- Enable easy code transfer to the Microsoft Azure environment – for the needs of managing a long-term cloud strategy

## 4.2 CMS security mechanisms

The CMS must have the following security mechanisms built-in:

- Regular cybersecurity testing by an external provider " **3rd party penetration tests**" proof of testing must be published on the website of the CMS solution provider (principal).
- Varna hramba gesel "Hashed passwords".
- Support for "Oauth", OpenID and SAML login systems.
- The use of SSL encryption for the security of data transfer.
- Two-factor authentication can be configured.
- "Content Security Policy (CSP)" configuration option.
- Configuration option:
  - CORS (Cross-Origin Resource Sharing):
  - Strict-Transport-Security Header (HSTS)
  - Cross-site scripting Protection (X-XSS-Protection header).
- Possibility of protecting API interfaces (API authentication and authorization, Rate Limiting...).

## 5 Tasks of Lot 2 – Website development

Lot 2 presents the tasks to be performed by the provider for the development of the website defined in Lot 1 in the selected CMS tool (it must comply with the standards defined in the technical specification), integration, migration and automation, setting up environments and controls, security, and performing testing and deployment.

### 5.1 Establishment of a development environment (DEV, STAGE, PROD)

#### **Purpose of the step:**

To establish a stable, secure and repeatable **development environment** within the client's infrastructure, which enables continuous development, testing and integration of the code for the new SPIRIT website, in accordance with the architectural guidelines set out in Section 1.

#### **Tasks of the provider**

- Establishment of three separate environments: development (DEV), test/staging (STAGE) and production (PROD).
- Setting security rules (WAF, anti-bot protection, robots.txt - blocking of searchers).
- Setting up a CI/CD pipeline within the client's GitHub account
- Establishment of automated backups .

#### **Handouts**

- Environment Architecture Documentation (PDF).
- CI/CD configuration scheme (diagram + description).
- Set up access to all environments for the client with separate roles.

## 5.2 Implementation of CMS foundations (structure, content types / modules)

### Purpose of the step:

Establish a fully functioning CMS system base that implements all the decisions from Lot 1:

- information architecture,
- A taxonomic model.
- A catalog of content types
- editorial workflows,
- SEO/WCAG/LLM standards.

Step 2 represents the foundation on which the entire portal will be built. All activities must be verifiable, and must follow the Lot 1 traditions.

### Tasks of the provider

- Implementation of the CMS structure in accordance with the IA and taxonomic model from Phase 1.
- Creation and configuration of **all types of content / modules**, including metadata (SEO, LLM, WCAG).
  - News, events, tenders, competencies, publications, contact points, investment stories, thematic landing pages, etc.
- Establishment of a system of labels, categories, segments (taxonomy).
- BE prepares templates for shared use in different content sets.

### Handouts

- Report on all implemented data types/modules and CMS configuration.
- Description of all prepared templates (plan, scope)

## 5.3 Implementation of visual templates (frontend development)

### Purpose of the step:

Implement all the templates, components, and UI logic **validated in Lot 1** in the CMS.

Step 3 is critical for:

- user experience,
- consistent interpretation of the umbrella identity of SPIRIT,
- accessibility,

- technical SEO compliance,
- Core Web Vitals performance.

All implementations must be **100% compliant** with the Design System — no deviations or ad-hoc solutions.

### Tasks of the provider

The provider must carry out the following activities:

- Implementation of FE templates based on validated Design System and prototypes (Consideration of WCAG 2.2 AA in each UI element).
- Implementation of FE library components (UI element library, style guide).
- Implementation of global components (header, footer, navigation, breadcrumbs).
- Implementation of responsive design (for all defined tipping points - chapter here: Responsiveness).
- Implementation of "Page Builder" toolbox components for use in building websites.

### Handouts

- Implemented FE templates in CMS.
- Component UI library.
- Toolbox with a set of all components in the CMS accessible to editors
- Report on WCAG and CWV compliance of FE templates.
- Style Guide (PDF + link in Figma/Storybook).
- Created typical prototype websites with demonstration content

### Request:

Each template must be fully implemented in accordance with:

- I'm using the Figma prototype.
- hierarchy of components from the Design System,
- components must be implemented as reusable components,
- standardi WCAG 2.2 AA (contrast ratio, keyboard nav, ARIA),
- SEO & LLM smernicami (structured data, semantic headings, metadata placeholders),
- performance smernicami ( $LCP \leq 2,5$  s,  $CLS \leq 0,1$ ,  $INP \leq 200$  ms).

#### 5.3.1 Odzivnost (Responsive implementation)

The new website should be fully customizable for different devices and screen sizes (flexible design), including mobile phones, tablets and desktops.

The Contractor must test and ensure that it works on different platforms and browsers, at least:

- Google Chrome – latest stable version (desktop + mobile)
- Microsoft Edge - Latest stable version (desktop + mobile)
- Mozilla Firefox – Latest Stable Version (Desktop + Mobile)
- Safari (Apple) – latest stable version (macOS + mobile)
- Mobile browser on Android (say Chrome for Android or another Chromium-powered one) and Safari on iOS

The provider must ensure that the templates and components are fully responsive to at least the following breaking points:

<b>Mobile</b>	320px - 480px	Smartphones (portrait)
<b>Tablet</b>	481px - 768px	Tablets (portrait), small laptops
<b>Small Desktop</b>	769px - 1024px	Tablets (landscape), smaller laptops
<b>Large Desktop</b>	1025px and up	Desktop monitors, larger laptops

**Request:**

- No horizontal scroll bars are allowed.
- Interactions must be optimized for touch (touch targets min. 44 px).

### 5.3.2 Accessibility

The Contractor must ensure that the new website complies with the Website and Mobile Application Accessibility Act (ZDSMA), **meets the requirements of WCAG 2.2 Level AA compliance, including a disproportionate burden assessment statement (inaccessible content on the website) and an accessibility statement.**

Web accessibility is key to ensuring that all users, including persons with disabilities and those with reduced abilities, can access online content and services. To ensure accessibility, it is important to follow both the technical guidelines as well as the legislative requirements and instructions of the UX team.

The provider must provide:

- correct ARIA tags,
- focus indicators,
- visual navigation with the keyboard,
- the correct hierarchy of addresses,
- alternative descriptions,
- correct contrast ratio,
- check the performance of screen readers (NVDA or VoiceOver).



### 5.3.3 SEO, LLM in Core Web Vitals optimizacija

When implementing a CMS solution, the contractor must comply with the following guidelines and requirements:

#### 1. Accessibility and indexation

- a. The URL structure should be clean, readable, and unique.
- b. An XML sitemap (sitemap.xml) must be in place *and the robots.txt must be set up correctly*.
- c. Canonical tags (<link rel="canonical">) should be used to manage duplicate content.
- d. Internal links must form a transparent hierarchy and allow access to all key pages in no more than three clicks.
- e. The CMS should be able to easily manage redirects for individual pages, posts or URLs. The administrator must be able to set up permanent (301) or temporary (302) redirects through the user interface.

#### 2. Architecture and upload speed (Core Web Vitals)

- a. The structure of the site must follow a logical hierarchy (home page → category → subpage).
- b. The website must score at least 90% on all criteria on the Lighthouse speed test. (Speed, SEO, Accessibility, Best Practices)
- c. Lazy loading mechanisms **for images and videos** should be implemented.
- d. All elements must be fully responsive (mobile-first).

#### 3. Structured data and metadata

- a. Each page must have neat **meta tags** (title, description, language, Open Graph/Twitter Card).
- b. H1-H3 headings must follow a logical content hierarchy.

#### 4. International and multilingual support

- a. Language versions must be clearly separated (e.g. /en/, /en/, /de/).
- b. Hreflang **tags must be implemented** to correctly identify language and regional variants.
- c. Automatic redirection based on IP or browser language is prohibited.

#### 5. Reliability and safety

- a. All pages must be accessible via **HTTPS**.
- b. The use of mixed content is not allowed.
- c. 301 redirects for removed or moved pages must work properly.

#### 6. LLM and generative optimization (e.g. for ChatGPT, Claude, Grok)

- a. Structured content must be prepared in such a way that it allows recognition in language models (Q&A, FAQ, HowTo structures).

- b. All pages should contain clear titles, replies and labeled entities (author, date, source).
- c. SEO guidelines should include **optimization for LLM models** in accordance with *the Generative Search Optimization (GMO)* guidelines.

The provider must implement:

- meta templates for all types of content,
- canonical,
- hreflang,
- JSON-LD strukture (schema.org),
- semantic headings (H1–H3),
- image optimization (WebP/AVIF),
- lazy-loading,
- overloading critical resources,
- Minimizing CSS/JS bundles.
- robots.txt and sitemap preparation.

Special requirement: Templates must be optimized for LLM models (ChatGPT, Claude, Grok) based on GMO (Generative Search Optimization) guidelines:

- Q&A markup,
- FAQ block,
- clean paragraphs and entities (organization name, authors, dates),
- structured content for AI summaries.

### 5.3.4 Analytics

In the phase of transferring the solution to the production environment, the client will take care of opening the "Google Analytics GA4" and "Google Search Console" accounts and provide the contractor with appropriate access.

The contractor must take care of the appropriate implementation of GA4 and Google tag manager in accordance with the UX instructions, which will allow to monitor the effectiveness of the SEO strategy and identify opportunities for improvement.

## 5.4 Backend implementation and integrations

### Purpose of the step:

Implement the entire backend logic, processes, integrations, validations, security mechanisms and data flows as defined in the Functionality Catalog, editorial flows, GDPR/ZVOP-2 requirements, SEO/LLM specifications and technical architecture.

The backend should be able to:

- stable implementation of all functionalities,

- protection of personal data,
- consistent behavior of the application,
- compliance with editorial workflows,
- reliable integrations,
- a comprehensive review of changes.

### 5.4.1 Implementation of integrations

Within the technical consolidation project, the client will unify the method of connecting applications and the method of obtaining data. To retrieve and store data in the CMS system, the so-called Spirit API HUB will be available

#### 5.4.1.1 *Spirit API HUB*

##### **Data retrieval**

For the purposes of data acquisition, API interfaces will be available within the Spirit API HUB platform, which will enable easy and secure access to resources such as external news, events, tenders.

In the offer, the Provider evaluates the creation of 3 Data Acquisition Integrations, taking into account the following items:

- Data retrieval using API REST calls with prior authorization and a known method described in the swagger documentation example: [Swagger UI](#)
- Saving data in CMS
- Display of retrieved data in the CMS and on the agency's website

For the purpose of storing data in the back-end CRM system, the provider prepares an offer for 2 integrations within the tender:

1. Saving event login data using CMS forms - using the CMS webhook for the created form in the CMS and calling a known API interface
2. Saving data for subscription to e-News - using the CMS webhook to create a form in the CMS and call to a known API interface

#### 5.4.1.2 *My SPIRIT*

As part of the offer, the contractor prepares an estimate of the work for the development of the My SPIRIT web interface, which must enable the following functionalities in the first phase:

- User registration (the user must be saved in the CMS solution as a user type without access to the editorial interface when registering)
- User login to My SPIRIT
- Possibility of logging in using SI-PASS

- Overview and ability to save preferences for displaying (personalized) content and notifications
- List of applications for SPIRIT calls – link to the JRP platform
- Overview of membership in the records of professional participation
- Possibility to register for economic delegations and to check the information in this regard and to print out the certificate of participation
- Possibility to add additional content elements related to the user profile
- Review your profile's personal information
- Possibility to change personal data
- Password Replacement

### **Tasks of the provider**

- Integrations with external systems according to the design and specification of the Spirit API Hub (CRM, analytics...).
- Implementacija API varnostnih pravil (rate limiting, input sanitization, encoding).
- Implementation of the My Spirit environment according to the given specification
- Ensuring compliance with GDPR and ZVOP-2 in all data flows.
- Implementation of email services (transactional mail, newsletter integrations).
- Implementation of logging and audit trail mechanisms

### **Handouts**

- Implemented all integrations according to the specification.
- Implemented all functional areas within My Spirit
- Integration documentation and data flow schema (PDF).
- Backend Service and Compliance Security Testing Report.

### **5.4.2 Compliance**

Compliance with industry regulations and legislation: The solution must comply with laws and regulations (ZVOP-2, GDPR, ZInFV-1 and WCAG 2.2 level AA) relating to data protection and user privacy.

The contractor must ensure that the new website is protected from the most common security vulnerabilities as defined in the OWASP Top Ten.

### 5.4.3 Cookie banner

The new website must comply with the General Data Protection Regulation (GDPR and ZVOP-2) and other relevant data protection laws.

This includes putting in place an effective cookie consent mechanism and ensuring that all data collection, processing, and storage practices comply with the requirements of the GDPR.

Requirements:

- The subscriber wants a cookie consent banner that appears when the user visits the website for the first time and provides clear information on the use of cookies and allows users to give or reject consent.
- The solution should allow users to customize their cookie settings, whereby they can opt for different categories of cookies (e.g. necessary, functional, analytical, marketing).
- It is necessary to provide users with an easy way to withdraw their consent at any time and to ensure that their preferences are immediately taken into account and enforced.
- It is necessary to provide a dedicated cookie policy page that provides detailed information about the types of cookies, their purpose and how users can manage their cookie settings.

## 5.5 Content migration – automated migration (contractor) and manual migration (UX agency)

### **Purpose of the step:**

To ensure a consistent, complete, verifiable and technically optimized transfer of all content from the four existing websites to the new unified SPIRIT website, in accordance with the Migration Playbook, Redirect Map, IA, taxonomy, SEO/WCAG/LLM standards and editorial models from Lot 1.

### **Tasks of the Contractor Provider Lot 2 – DEVELOPMENT**

The provider must carry out *the purely technical part of the migration*, which includes:

#### 5.5.1 Preparation of automated migration flows according to the Migration Playbook

The provider must:

- Set up a migration tool or scripts.
- implement content extraction from old URLs (supported sites: spiritslovenia.si, izvoznookno.si, podjetniski-portal.si, sloveniabusines.eu),

- implement data transformation into CMS content type structures (100% match with fields),
- implement category mapping logic in a new taxonomy,
- Implement URL transformation logic → new semantic URLs
- check encoding, images, attached PDF documents, attachments.

#### **Request:**

Migration scripts must be able to:

- *replicable implementation, if necessary,*
- *partial re-migration;*

### 5.5.2 Implementation of Redirects (301 Redirect Map)

The provider must:

- implement **the entire Redirect Folder** validated in Section 1,
- Ensure that 301 redirects:
  - Work for 100% of all old URLs that are defined.
  - return the correct codes (301, not 302),
  - They are crawler tested.

### 5.5.3 Prepare your environment and tools for manual migration

The provider must provide the UX to the agency:

- editorial rights and access to the CMS,
- input forms consistent with the content typology,
- functioning media content management,
- validation messages,
- Page templates ready for manual input
- Instructions for work.

#### **Request:**

The provider must provide **technical support to the UX agency** at the time of manual migration:

- debugging,
- fixing scripts,
- Help with data issues.

### 5.5.4 Final validation of the total migration volume

After completing the manual migration (performed by the UX agency), the provider must perform:

- checking the structure for SEO (title, H1, meta, canonical),
- WCAG (alt text, title structure) checking,
- checking compliance with the Redirect Folder,
- checking the functionality of sheet pages,

The provider does NOT check the correctness of the content (this is the responsibility of the client), but **the correctness of the display and compliance with technical standards**.

## **Handouts of the provider**

### **1. Automation scripts and documentation**

The document must contain:

- a description of the migration architecture;
- Tools used
- a description of all mapping fields (mapping table),
- A list of errors and exception handling.

### **2. Automated migration report (PDF)**

It includes:

- A table of all migrated content (numeric)
- % of automated migration,
- % of failed records and reasons,
- revised data.

### **3. Implemented Redirect Folder (final)**

- in a CMS or server configuration,
- tested and confirmed by the client.

### **4. Open and closed bug report (JIRA or PDF)**

- class of error,
- status,
- responsible person,
- Date of resolution.

## 6 Testing (QA, WCAG, SEO, performance and security checks) with demarcation of competence

### **Purpose of step:**

Ensure that the system is technically, substantively, visually and functionally compliant with the requirements of Lot 1 and Lot 2.

Testing is carried out by **two teams** with different responsibilities:

- **Development team (Lot 2 contractor)** → technical and system testing
- **UX agency (Lot 1 contractor)** → user experience, UX content review, visual compliance

### **Tasks of the development team**

- Functional testing of all modules.
- SEO technical review + LLM indexing.
- Core Web Vitals testing on the STAGE environment.
- WCAG 2.2 AA Review and Corrections.- Technical Testing
- Security testing (OWASP ASVS: input, auth, session, data).
- Load testing (at least 2000 concurrent visits).
- Test of automated workflows (forms, emails).

### **Handouts**

- QA Test Report (PDF).
- WCAG Audit Report (PDF).
- SEO & LLM Compliance Report (PDF).
- CWV Performance Report (PageSpeed Insights + Lighthouse).
- Common patch matrix (JIRA/Excel) with deadlines and responsibilities for editing corrections



## 7 Introduction to production (Go-Live)

### Purpose of the step:

Perform a controlled, documented and secure deployment of the entire solution into the production environment, including the migration of the final data set, the last technical checks, configurations, SEO/WCAG validations, and ensuring the stability of the system in the first days after launch.

### 7.1 Tasks of the DEVELOPMENT TEAM

The provider must carry out all technical activities related to the launch.

#### Tasks of the provider

- Final review of all templates and functionalities.
- Migration of the final dataset to production.
- Preparation of the Go-Live plan (schedule, security procedures, rollback).
- Establishment of production settings (**GA4 analytics**, SEO, robots, sitemaps).
- On-call support.

#### Handouts

- Go-Live Runbook (PDF).
- Production deploy log.
- Report on the successful establishment of sitemaps, robots.txt, CWV initial metrics.

#### 7.1.1 Preparation and approval of the Go-Live plan (Runbook)

The provider must prepare a **Go-Live Runbook** that includes at least:

- a list of all activities before, during, and after launch;
- responsible persons (development, UX, client),
- Implementation timeline (timeline with precise steps)
- checking dependencies (DNS, servers, SSL, APIs),
- checking backups and rollback process,
- manual or automated deployment process,
- a list of critical points and risks,
- a plan of communication with the client,
- Instructions for escalating incidents.

#### Request:

The runbook must be confirmed **at least 5 business days** before Go-Live.

### 7.1.2 Go-Live deploy

The provider is responsible for:

- implementation of production deployment,
- checking the operation of key functionalities after deployment,
- Write logs to the Runbook.
- Fix any issues in real time.

### 7.1.3 Technical support at the time of launch

The provider provides:

- **on-call support** for at least 24 hours from the moment of launch,
- quick response to critical errors (up to 30 minutes),
- Elimination of blocker errors on the same day (if possible).

## 7.2 Tasks of the client

The Client is responsible for:

- Runbook confirmation
- confirmation of migrated content (content part),
- implementation of business QA validation (legal compliance, communication requirements),
- Confirmation of the Go-Live moment.

## 7.3 Post-Go-Live Stabilization (Post-Go-Live)

Within a period of 5 working days after Go-Live, **the** Lot 2 provider must:

- monitoring logs,
- elimination of errors that occur in production,
- making corrections for minor visual deviations,
- additional performance optimizations (if necessary),
- validation of Core Web Vitals metrics (the first real-user signal is collected for several days),
- reporting to the client.

## 8 Final documentation, training and handover of the system

### **Purpose of the step:**

To ensure that the client takes full technical and editorial control over the system, receives all technical documentation, structured editorial documentation, conducts trainings and receives formally validated final handouts.

This step represents **the official handover of the project** and confirms that the system is ready for long-term management.

## 8.1 Preparation of complete technical documentation

The tenderer must prepare **a complete technical documentation**, which must be at least 80-120 pages and must include:

- System architecture
- CMS implementation
- Modules and functionalities
- Integrations.
- CI/CD
- Security mechanisms.
- Backup in monitoring

## 8.2 Preparation of editorial documentation

*(implemented by the UX agency, but the developer provides access to the CMS for technical snapshots of the procedures)*

## 8.3 Training for editors (DEVELOPMENT AND UX TEAM)

**The Phase 2 Contractor + UX agency is carried out jointly.**

Division of responsibilities:

- **UX agency:** content and UX use of CMS (proper writing, content structure, editorial rules)
- **developer:** technical use of CMS (content types, workflows, taxonomy, search engine, forms)

The provider must carry out:

- **at least 2 separate training modules**, 2-3 hours each:
  1. Create and edit content, including accessibility rules
  2. Taxonomy, translation and workflow management
  3. Use of modules (calendar, publications, news, calls, contacts...)
- Training must be:
  1. live (online or onsite),
  2. with the material,
  3. filmed.

## 8.4 Technical training for the client's IT team (DEVELOPMENT TEAM)

The provider must carry out **at least 1 technical module**, which includes:

- the architecture of the system,
- CI/CD pipeline,
- backup/restore pipeline,
- management of environments,
- monitoring,
- integration,
- security configurations,
- troubleshooting.

## 8.5 Final validation and signature record

The provider must carry out:

- presentation of the entire solution,
- demonstration of the main functionalities,
- handing over documentation,
- the elimination of the last possible deficiencies,
- Signature record of the handover of the system.

# 9 Download the code

The contractor must transfer all source code developed within this project, including the entire frontend and backend, to the client's Spirit Github account no later than after the completion of the project (after signing the transceiver for the transition to the live environment) and update the code on Github with each change.

The tenderer must submit:

- the entire code repository,
- configuration of CI/CD pipelines,
- Access Points Document,
- list of service accounts,
- a list of locations where API keys are stored (without exposing the key),
- Instructions for restoring the environment.

# 10 UX and Development Team Collaboration

To ensure consistency between the design design, technical execution and cost-effectiveness of the project, the UX/UI and the development team must work closely together in all phases of the project.

### Consideration of the technological capabilities and limitations of the CMS

- The UX design must be based on the functional capabilities of the selected CMS system.
- When planning, the UX team must take into account the limitations of the chosen CMS platform – especially in terms of content structure, components, modules, and management of multilingualism and workflows.
- When designing user experience and functionality, it is necessary to use the existing ("off-the-shelf") solutions of the CMS system as a priority.
- Custom development is only permissible in cases where:
  - the existing solution does not meet the functional or user requirements,
  - Custom development brings significant added value to the user experience or business process.
- Each custom development proposal must be justified in writing (description of the need and impact on UX) and previously approved by the client.

### Collaboration between the UX design and development phase

- During the design phase, the UX team must check the feasibility of individual elements in cooperation with the development team.
- Cooperation should be **targeted and effective**, focusing on:
  - harmonisation of specifications;
  - explanation of interaction patterns and components,
  - preparation of *redlines* documentation,
  - clarifications on user workflows, roles and rights.
- All communications and agreements must be documented and accessible in a common project repository

The goal of such cooperation is to ensure that the UX design:

- supports realistically feasible solutions within the selected CMS,
- enables optimized development without unnecessary adjustments,
- maintains a high level of user experience while rationally spending development resources.

## 11 Roles and responsibilities of the project team

**Naročnik (SPIRIT Slovenija):**

- provides access to stakeholders, editors and relevant internal sources;
- provides CGP and brand materials, brand hierarchy and existing content bases,
- provides timely clarifications and decisions necessary for the smooth work of the Agency;
- confirms key decisions on project milestones (e.g. IA, UX concept, UI design, CMS architecture),
- coordinates the involvement of internal departments (legal department, PR, translations).

**Stakeholders (legal department, marketing, translation team):**

- participate in the substantive harmonization and verification of the compliance of publications,
- confirm terminology, legal correctness, compliance with the mark and linguistic consistency,
- participate in the preparation of the final content for migration.

**The UX team:**

- manages the user experience design process (analysis, IA, wireframes, prototypes, testing),
- co-creates *to-be* processes with the client and editorial teams,
- prepares UX/UI documentation, design system and interactive prototypes,
- participates in workshops and tests with editors and stakeholders,
- Works purposefully with the development team for feasibility checks and clarifications.

**Development Team:**

- participates in the validation of the feasibility of individual elements and components,
- takes over the design system, UX/UI prototypes and specifications as an input to development,
- Requires explanations from the UX team regarding interactions, workflows, and technical details (as needed).
- He takes care of technical implementation, integration, testing and handover of solutions.

## 12 Key project milestones and payment dynamics

Milestone	Timeline	Handout development	Payment Development
<b>M1 - Strategic and user design validation</b>	T12	<ul style="list-style-type: none"> <li>• Technical confirmation of the feasibility of UX solutions</li> <li>• Draft CMS data model (content types, relationships)</li> <li>• Draft FE architecture (component approach)</li> <li>• Development assessment and sprint plan</li> </ul>	5%
<b>M2 – Design System &amp; Prototypes + Start of Development</b>	T16	<ul style="list-style-type: none"> <li>• DEV environment</li> <li>• CMS core (content types, taxonomy)</li> <li>• Global elements (header, footer, navigation)</li> <li>• Documentation of environment architecture</li> </ul>	5%
<b>M3 – By UX testing: Prototypes v2</b>	T20	<ul style="list-style-type: none"> <li>• FE Template Set: Implemented FE templates in CMS. Component UI library. Toolbox with a set of all components in the CMS accessible to editors</li> </ul>	20%
<b>M4 – Template Set 3 + Integracije</b>	T24	<ul style="list-style-type: none"> <li>• Implemented CMS foundations (content types, taxonomy, workflows)</li> <li>• Implemented all FE templates and components with demo content</li> <li>• Integrations (Spirit API HUB)</li> <li>• My SPIRIT – MVP</li> <li>• WCAG &amp; CWV, GDPR FE compliance report</li> <li>• Technical documentation v1</li> </ul>	20%
<b>M5 – Pilot migration to STAGE</b>	T28	<ul style="list-style-type: none"> <li>• STAGE environment</li> <li>• Pilot migration of content</li> <li>• QA report v1 (functional, WCAG, SEO, LLM)</li> <li>• List of errors and corrections with timeline of elimination</li> </ul>	10%
<b>M6 – Final Migration + QA v2</b>	T32	<ul style="list-style-type: none"> <li>• Final automated migration</li> <li>• 301 Redirect Map (implemented and tested)</li> <li>• SEO, WCAG 2.2 AA, LLM, CWV reports v2</li> <li>• Go-Live Runbook</li> <li>• Technical documentation v2</li> </ul>	5%
<b>M7 – GO LIVE Production deploy + stabilization</b>	T36	<ul style="list-style-type: none"> <li>• Production deployment</li> <li>• Stabilization (bugfixes, monitoring)</li> <li>• First production SEO/CWV data</li> </ul>	10%

<b>M8 – Final handover of the system</b>	T39	<ul style="list-style-type: none"> <li>• Submission of code, repositories and accesses</li> <li>• Technical documentation (final)</li> <li>• Training of the IT team</li> <li>• Signing of the acceptance record</li> </ul>	5%
--	-----	---	----

20 % of the contract value shall be retained and paid after the lapse of two (2) months from the successful transition of the system to the production environment, provided that all identified defects have been rectified within that period.

## 12.1 Explanations

### 12.1.1 Definition of a completed milestone

A milestone shall be considered **completed** when all of the following conditions are cumulatively met:

1. All handouts defined for each milestone in the tender and contract documentation are fully produced and handed over to the client in the agreed form and scope.
2. The handouts comply with the requirements of the tender documentation, technical specifications, UX specifications and relevant standards (WCAG 2.2 AA, SEO, GDPR, ZVOP-2, Core Web Vitals, LLM optimization).
3. The contractor has provided the client with a written declaration of conformity confirming that all deliveries are made in accordance with the contractual obligations.
4. The contractor has corrected all critical (P1) and major (P2) nonconformities identified in the handout review process.
5. The submissions have been examined and confirmed in writing as appropriate by the client or its authorised independent expert.

### 12.1.2 Milestone review and validation process

1. The Contractor shall inform the Client in writing that the handouts of each milestone are ready for inspection and shall at the same time submit all the relevant handouts and documentation.
2. The client has at least 7 working days from the receipt of the handouts to review and make comments.
3. If the Client does not submit comments within the specified period, the handouts are considered conditionally confirmed, unless the contract stipulates otherwise.
4. If the Client submits comments, the Contractor must eliminate all reasoned comments (critical (P1) and major (P2) non-conformities) without additional costs within the deadline set by the Client, and usually not longer than 7



working days, unless it is a matter of more extensive changes agreed upon by the parties in writing.

5. Once the objections have been resolved, the contractor shall resubmit the handouts for approval.

### 12.1.3 Criteria for confirming a milestone

A milestone shall be confirmed when the following criteria are met:

- functionalities operate in accordance with defined specifications,
- the handouts do not contain critical or major errors,
- any minor deficiencies (P3) are recorded in the list of open points with an agreed deadline for elimination,
- Handouts allow the next phase of the project to proceed smoothly without additional dependencies or risks.

### 12.1.4 Milestone-to-payment ratio

1. Payment for each milestone is due only after its written confirmation by the client.
2. The Contracting Authority reserves the right to withhold the payment or its proportionate part until all identified non-conformities have been rectified.
3. The payment of the milestone does not mean the final takeover of the entire project, nor does it limit the rights of the contracting authority under the warranty or liability of the contractor.

### 12.1.5 The role of the independent expert

Where the contracting authority appoints an independent expert or project manager:

- its written assessment of the conformity of the deliveries shall be considered as the professional basis for the contracting authority's decision to confirm the milestone,
- the contractor is obliged to provide an independent expert with access to handouts, drafts and technical documentation,
- The comments of the independent expert shall be considered as comments of the contracting authority.

### 12.1.6 Moving to the next milestone

A transition to the next milestone or phase of a project shall be permitted even before the written confirmation of the previous milestone, provided that the identified comments or deficiencies do not affect the implementation of the activities of the next milestone and do not pose a critical risk to the quality, compliance or further progress of the project.

In such a case:

1. The deficiencies identified are recorded in a list of open points, with clearly defined deadlines for their rectification.
2. The contractor is obliged to eliminate the deficiencies within the framework of the ongoing project and at no additional cost, within the agreed deadline.
3. The continuation of activities in the next milestone does not constitute confirmation of the previous milestone, nor does it affect the right of the contracting authority to request the elimination of identified non-conformities.
4. Payment for the previous milestone is due only after its written confirmation, regardless of the fact that work is already carried out in the next milestone.
5. If it turns out below that the identified deficiencies nevertheless affect the activities of the next milestone, the contracting authority has the right to request appropriate corrective measures or adjustment of the timeline at no additional cost.

#### **A. Examples of what it means that identified deficiencies *do not affect* the activities of the next milestone**

The identified deficiencies **shall not affect the implementation of the activities of the next milestone** where all of the following conditions are met:

##### **1. They don't affect the technical architecture or data model**

Examples:

- minor visual corrections (color contrasts, spacing, typographic corrections) are missing,
- inconsistencies in the naming of AI elements,
- minor corrections to navigation tags (labeling),
- Descriptions are missing or incomplete in the documentation.

**They affect:**

- Changes to content types
- changes in relationships between contents,
- changes to the taxonomy;
- changes to the basic navigational structure.

##### **2. Do not affect user paths (JTBD)**

Examples:

- minor UX tweaks within already confirmed user paths,
- improvements in microinteractions,
- corrections to CTA texts without changing the logic of the path.

**They affect:**

- Add or remove steps in a JTBD path.
- change the sequence of steps,
- change in user journey goals.

### **3. They do not affect development activities that are already planned**

Examples:

- UX fixes that can be implemented within existing FE components,
- changes that do not require a change in backend logic,
- Corrections that can be included in an ongoing sprint without changing the scope.

**They affect:**

- changes that require code refactoring,
- modifications requiring new FE/BE components,
- changes that require a change in the sprint plan or a delay in deadlines.

### **4. Do not affect compliance with standards**

Examples:

- Minor WCAG improvements that don't affect the underlying structure (e.g., alt texts, contrast fixes)
- Complementing SEO metadata or structured tags.

They affect:

- non-compliance with WCAG 2.2 AA,
- Errors that cause the page to not be indexed.
- violation of the GDPR or ZVOP-2.

## **B. Examples of substantiated objections that the contractor must remedy at no additional cost**

Justified objections are those arising from **non-performance of contractual obligations** and do not constitute a change in scope.

### **1. Non-compliance with the tender or contract documents**

Examples:

- Handout doesn't include all required items (e.g., certain content types are missing)
- the functionality is not implemented as described in the offer,
- Mandatory handouts (reports, documentation) are missing.

## **2. Non-compliance with certified UX specifications**

Examples:

- The FE implementation does not follow validated prototypes or the Design System.
- Wrong content or navigation hierarchy
- misuse of components.

## **3. Non-compliance with standards**

Examples:

- inadequate compliance with WCAG 2.2 AA,
- technical SEO does not meet the agreed criteria (e.g. CWV "Good" threshold),
- mandatory SEO/LLM structural data is missing,
- failure to comply with GDPR or ZVOP-2 requirements.

## **4. Functional errors**

Examples:

- the functionality does not work according to the specification,
- errors that make editorial work impossible,
- Errors that cause misrepresentation or loss of data.

## **5. Migration errors**

Examples:

- incorrectly mapped content according to Mapping Matrix,
- Missing or incorrect 301 redirects
- Disconnections above the threshold (<1%).

## **6. Insufficient documentation or education**

Examples:

- Steps are missing in the editor's manual.
- the documentation is not clear enough for independent use,
- The education did not cover all the agreed content.

**What is NOT a valid objection (and is treated as a change in scope):**

- new requirements that were not part of the specification,
- additional functionality or integration,

- changes to business rules,
- aesthetic preferences that are not part of the certified UX,
- changes in legislation following the confirmation of the milestone (if not explicitly covered by the contract).

## 13 Maintenance

### 13.1 Warranty during the warranty period

After the completion of the takeover of the established solution into production operation, warranty maintenance will follow for a period of twelve (12) months from the date of signing the acceptance record on the establishment of the website.

The Contractor guarantees that the site will work in accordance with the specified requirements and instructions for use, otherwise it will eliminate all defects free of charge during the warranty period.

An error is defined as a non-functioning of the website or operation that does not comply with the requirements specified in the tender documentation and those requirements that are subsequently agreed with the contractor in writing.

### 13.2 Maintenance

The maintenance contract is concluded for a period of 60 months from the date of signing the handover record on the production establishment of the solution.

Maintenance is divided into:

1. Routine Technical Maintenance
2. Redna podpora (Support)
3. Additional work and development (Change Requests)

#### **Performing maintenance (applies to all three types of maintenance)**

- The Contractor undertakes to provide technical support to editors and maintenance during regular working hours (as a rule, from Monday to Friday from 8:00 a.m. to 4:00 p.m., all working days in the Republic of Slovenia).
- For the purpose of managing claims, the contractor will offer a suitable technical solution through which the client will be able to monitor the status of the case, the history of correspondence, etc. at any time (the so-called ticketing system).
- Upgrades and maintenance works, which result in the temporary unavailability of web portals, are carried out at night or over the weekend.
- The contractor undertakes to announce any major interventions of upgrades or installation of safety patches one week in advance to the agreed e-mail address and to agree with the client on the most appropriate time for the implementation of the interventions.
- Exceptions are critical security patches that are installed by the contractor at the first appropriate opportunity within the required time window (no later than 5 days after the patch is published).

- The contractor takes care of all administrative and administrative tasks related to the implementation of the contract.
- With each refinement and/or change, the contractor ensures that the system documentation (technical and user) is updated.
- For any unjustified and unagreed delays (either in the elimination of errors in accordance with the SLA or in the implementation of additional works in accordance with each offer), the client reserves the right to impose a contractual penalty as specified in the model contract.

### 13.2.1 Routine technical maintenance, covering in particular:

- 1x annual updating of the CMS to the latest supported version (minor versions) by prior agreement with the client;
- To install CMS security patches in real time:
  - o Fixes for critical vulnerabilities no later than 5 days after publication,
  - o Other security patches no later than 15 days after publication;
- 1x annual security review of code and configuration;
- 2x annual verification of Core Web Vitals (e.g. with Google Lighthouse, PageSpeed Insights, Web Vitals extension...) – all three CWV indicators must be within the recommendations:
  - o  $LCP \leq 2.5\text{ s}$ ,
  - o  $INP/FID \leq 100\text{ ms}$ ,
  - o  $CLS \leq 0.1$ .
- Monitoring system alerts and informing the client about the necessary actions;
- Maintenance of a dedicated database;
- Trivial/minor adjustments due to changes in legislation (e.g. GDPR/ZVOP-2 structure);
- Kvartalni “health check” sistema.

The service is billed as a monthly lump sum.

### 13.2.2 Regular Support, which includes, in particular:

- Resolution of technical incidents (P1, P2, P3);
- Assistance in resolving CloudFlare incidents;
- Assisting editors in using CMS (minor issues);
- Assisting the IT team with configurations (CI/CD, environments);
- Elimination of minor defects after the expiration of the warranty period;
- Assistance with editorial activities (minor corrections to content types, AI deviations);
- Review of new content in the context of SEO/LLM (light improvements);
- Assisting in updating editorial roles, taxonomy, or petty processes;
- Other similar works that fall within the field of assistance and counseling.

### SLA – Debugging

Type of error	Response time	Rescue time	Example
Q1: Critical errors	1h	1-2h, continuous to solution	The website is completely unresponsive, access to the CMS is disabled; Critical functionalities not working
P2: Srednje	2h	up to 72h	The website is basically responsive, but the operation is slow, non-critical functionalities are detected, etc.
P3: Everything else	1 day	By agreement with the client, as a rule, no more than 7 days	

### SLA – Help/Advice

Response to request: within 1 day  
Implementation: by agreement with the client, usually within 7 days at the latest

### Method of implementation of support

- The type of error is determined by the client at the time of registration at his own discretion based on the above criteria through the ticketing system.
- The response time is the time that elapses from the receipt of the client's application to the acceptance by the contractor.
- When solving a critical error, if it cannot be eliminated within the prescribed period, it must provide a temporary solution (the so-called workaround) within the prescribed period, after which the error is eliminated continuously until it is completely eliminated.
- Reporting a critical jam is also possible by phone.
- If the Contractor assesses that the delay or defect is such that it will not be possible to eliminate it within the required time, it shall immediately notify the Client in writing, with an explanation and a proposal for the time within which the delay can be eliminated. The contracting authority shall confirm or reject the proposal in writing to the contractor. A rejected proposal must be explained by the contracting authority.
- An error can be reported by the contact person of the client or contractor.
- If it is a matter of eliminating a defect in the infrastructure, the defect must be eliminated by the contracting authority.



The client estimates that they will need an average of 10 hours per month of support in the first year, and an average of 5 hours per month in subsequent years. The service is charged as a monthly lump sum for the specified volume. Unused hours and any higher consumption are carried over to the following months, once a year in the case of higher consumption than planned, a settlement is made with additional payment (i.e. for hours over 120 hours in the first year and for hours over 60 hours in subsequent years).

The flat rate is 60 minutes, the billing interval is 15 minutes.

The contractor must submit a report on the interventions performed and the hours of support actually performed in the monthly maintenance bill. In the event that the contractor does not attach the report, the client will reject the invoice.

### 13.2.3 Upgrades (Change Requests), which includes, for example:

- Development of new functionalities, in accordance with the perceived needs;
- New sections or new content types;
- Search engine extensions;
- Integration with new APIs;
- Automation of editorial processes;
- Improvements to LLM optimization (AI search appropriate);
- Performance optimization → CWV changes;
- Additional templates (landing pages);
- Preparation of mini-campaign templates;
- Migrations from external systems.

#### Method of carrying out additional work

- The client will forward any additional functional requirements or upgrades through a dedicated system (the so-called ticketing system) to the contractor.
- In the inquiry, the contracting authority will define the subject of the order, the content and propose the deadline for implementation.
- The contractor's offer must show the specification of services, the range of hours required and the deadline for the performance of services.
- When sending an offer of additional services or upgrades, the contractor must take into account the price of the hourly rate offered under this public contract.
- After the offer is confirmed by the client, the contractor develops the solution and places it on the client's test environment.
- After the client confirms the suitability of the implemented solution, the contractor moves the solution from the test to the production environment and prepares a handover record, which is the basis for issuing an invoice.