**Příloha č. 10 Zadávací dokumentace:**

**ZPŮSOB HODNOCENÍ**

**KVALITY ZDROJOVÉHO KÓDU VYBRANÉ ZAKÁZKY**

|  |  |
| --- | --- |
| **Název veřejné zakázky** | **Pořízení nových webových stránek III** |
| Druh řízení | výběrové řízení veřejné zakázky malého rozsahu |

|  |  |
| --- | --- |
| **Zadavatel:** | **Město Slatiňany** |
| sídlo**:** | T. G. Masaryka 36, Slatiňany, PSČ 538 21 |
| IČ: | 00270920 |
| DIČ: | CZ00270920 |
| zástupce: | MVDr. Ivan Jeník, starosta |

|  |
| --- |
| IDENTIFIKAČNÍ ÚDAJE DODAVATELE (ÚČASTNÍKA): |
| **Obchodní firma / název:** | [doplní dodavatel] |
| sídlo: | [doplní dodavatel] |
| IČ: | [doplní dodavatel] |
| DIČ: | [doplní dodavatel] |
| zástupce: | [doplní dodavatel] |

(dále jen **„dodavatel“**)

Kvalitu zdrojového kódu vybrané zakázky v rámci každého jednotlivého hodnotícího subkritéria a každé jednotlivé hodnocené vlastnosti zdrojového kódu bude posuzovat zadavatel či příslušná hodnotící komise

V rámci posouzení a hodnocení kvality zdrojového kódu vybrané zakázky bude zadavatel či hodnotící komise vycházet z odborných znalostí, zkušeností a dovedností osob pověřených hodnocením i podkladů předaných účastníky v jejich nabídkách.

Posouzena a hodnocena bude každá jednotlivá položka nabídky samostatně, tj. bude zvlášť hodnocena každá jednotlivá vlastnost zdrojového kódu v rámci hodnotícího subkritéria, a to přidělením příslušného počtu bodů dle níže uvedeného „**Bodového ohodnocení vlastností zdrojového kódu“** (v rámci každé jednotlivé hodnocené vlastnosti zdrojového kódu). Hodnoceno bude v rámci bodové škály u jednotlivých vlastností (funkcionalit) zdrojového kódu následujícím způsobem:

* **2 body (zcela splněno),**
* **1 bod (částečně splněno)** *(pozn. u některých funkcionalit nemusí být hodnocení částečného splnění)*
* **0 bodů (nesplněno)**

**Výsledkem hodnocení kvality zdrojového kódu zakázky**, dle hodnocení každé jednotlivé vlastnosti zdrojového kódu v rámci hodnotících subkritérií, bude **součet** obdržených bodů v podobě **Váženého počtu bodů za kvalitu zdrojového kódu vybrané zakázky**, kdy maximální možný počet bodů, kterých může dodavatel dosáhnout, je **28 bodů**.

Zadavatel upozorňuje, že posouzení a hodnocení kvality zdrojového kódu zakázky je předmětem hodnocení v rámci ***hodnotícího kritéria č. 2 - Kvalita zdrojového kódu zakázky (s vahou 25%)***.

|  |
| --- |
| **ZPŮSOB HODNOCENÍ****KVALITY ZDROJOVÉHO KODU ZAKÁZKY** |
| **Pol. č.:** | **Název hodnoceného subkritéria v rámci hodnocení kvality zdrojového kódu** | **Stručný popis subkritéria v rámci hodnocení kvality zdrojového kódu** | **Hodnocená vlastnosti zdrojového kódu** | **Bodové ohodnocení****vlastností zdrojového kódu**  |
| **1.** | **Názvy proměnných, funkcí, tříd** | *Názvy proměnných by měly být určeny podle toho, jakou informaci v sobě nesou.* *Názvy funkcí naopak podle toho, co funkce provádí.* *Není vhodné používat zbytečně dlouhé názvy, a také se vyhnout používání jednopísmenných a nicneříkajících názvů.* | **Jazyk** | * ***skládání z angličtiny[[1]](#footnote-1)***
 | **2 bodů** |
| * ***jiný***
 | **0 bodů** |
| **Způsob skládání slov**  | * ***camelCase***
 | **2 bodů** |
| * ***pascalCase / snake\_case***
 | **1 bod** |
| * ***jiný***
 | **0 bodů** |
| **Vhodné pojmenování (tj. aby většina případů byla pojmenována vhodnými tematickými slovy o délce maximálně dvou výstižných slov bez spojek)** | * ***vhodné pojmenování, vhodná délka, tematické pojmenování slovy, konvence dle důležitosti slov při skládání***
 | **2 bodů** |
| * ***vhodné pojmenování, nevhodná/nesmyslná délka***
 | **1 bod** |
| * ***nevhodná / nesmyslná délka, nevhodné / nesmyslné pojmenování***
 | **0 bodů** |
| **2.** | **Logika funkcí** | *Posouzení zda funkce naplňují z hlediska obsahu požadovanou funkcionalitu a nedělají věci navíc. Kontrola délky funkcí. Očekává se, že funkce by měla dělat jen jednu věc a dělat ji řádně.* | **Obsah, stručnost a výstižnost funkcí** | * ***kód je čitelný a neobsahuje žádné obtížně pochopitelné funkce***
 | **2 bodů** |
| * ***kód je čitelný, až na výjimky neobsahuje dlouhé nesrozumitelné funkce***
 | **1 bod** |
| * ***tzv. „spaghetti code“, tj. nevhodná struktura zdrojového kódu, která způsobuje jeho nesrozumitelnost a obtížnou udržovatelnost***
 | **0 bodů** |
| **3.** | **Objektově orientovaný přístup** | *Kód vychází z objektového popisu řešeného problému, který je abstrakcí skutečného problému. Návrh poskytuje vhodnou strukturu pro řešení tohoto problému.* | **Konceptuální normalizace**  | * ***každý objekt má své jasné určení***
 | **2 body** |
| * ***většina objektů má své jasné určení, ale vyskytují se i objekty bez jasného určení***
 | **1 bod** |
| * + ***žádný objekt nemá své jasné určení***
 | **0 bodů** |
| **Využívání návrhových vzorů** | * ***ANO***
 | **2 body** |
| * ***NE***
 | **0 bodů** |
| **4.** | **Struktura v návaznosti na architekturu projektu** | *Vhodné vytvoření složek a souborů.* | **Využití správce artifaktů** *(např. composer, yarn, npm, apod.[[2]](#footnote-2))* | * ***použit***
 | **2 body** |
| * ***nepoužit***
 | **0 bodů** |
| **Využití frameworku** *(Symfony, Vue, React, apod.[[3]](#footnote-3))* | * ***použit veřejně dostupný***
 | **2 body** |
| * ***vlastní***
 | **1 bod** |
| * ***nepoužit***
 | **0 bodů** |
| **Architektura projektu** *(MVC, MVVC, apod.[[4]](#footnote-4))* | * ***použití standardizované architektury v plném rozsahu***
 | **2 body** |
| * ***použití standardizované architektury pouze z části***
 | **1 bod** |
| * ***použití jiné architektury (nestandardizované, vlastní apod.)***
 | **0 bodů** |
| **Způsob rozložení architektury v adresářové hiearchii, pojmenování souborů** | * ***dodržení standardu frameworku***
 | **2 body** |
| * ***přehledná hierarchie a pojmenování souborů, avšak nedodržující standard frameworku***
 | **1 bod** |
| * ***Nepřehledná hierarchie i pojmenování souborů***
 | **0 bodů** |
| **5.** | **Formátování kódu** | *Formátování je velmi důležitá věc. Pokud bychom psali kód bez bližšího uvážení nad jeho formátem, zhoršilo by to možnosti vyhledávání v textaci, tj. byl by obtížně či zcela nepoužitelný. Proto je důležité, abychom kód formátovali přehledně, stylisticky úhledně a ideálně s použitím automatického formátování, díky čemuž se v něm budeme i dobře orientovat.* | **Dodržování jednotného formátování** | * ***ukázkový kód používá nástroje na kontrolu automatického formátování***
 | **2 body** |
| * ***kód nepoužívá nástroj na automatické formátování, ale je formátován očividně ručně***
 | **1 bod** |
| * ***kód není formátovaný***
 | **0 bodů** |
| **6.** | **Komentáře v kódu** | *Komentáře jsou v informatice součásti syntaktických pravidel zápisu programu v mnoha programovacích jazycích. Komentář se vkládá před nebo za část zdrojového kódu (např. příkaz), jehož funkci nebo význam je potřeba vysvětlit. Používání komentáře není povinné, umožňuje však napomoci k pochopení činnosti programu.**Komentáře by měly být psané spíše sporadicky, protože samotný kód je jeden velký komentář. Kód píšeme tak, aby se dobře četl a nepotřeboval komentáře.* | **Komentáře funkcionalit zdrojového kódu**  | * ***Komentáře nejsou potřeba (zdrojový kód je zcela zřejmý), nebo je komentářů jen úměrné množství (do 10% funkcionalit)***
 | **2 body** |
| * ***10% - 50% funkcionalit není zřejmých a není u nich uveden ani komentář***
 | **1 bod** |
| * ***Více než 50% funkcionalit není zřejmých a není u nich uveden ani komentář***
 | **0 bodů** |
| **Kód využívá anotační způsob popisu metod a objektů** *(např. JavaDoc, PHPDoc, apod.[[5]](#footnote-5))* | * ***používá u všech veřejně přístupných objektů, metod a proměnných***
 | **2 body** |
| * ***používá u některých (části) veřejně přístupných objektů, metod a proměnných***
 | **1 bod** |
| * ***nepoužívá***
 | **0 bodů** |
| **7.** | **Testy** | *Čím vyšší je procentuální pokrytí kódu testy, tím je pravděpodobnější, že se budoucnu předejde zbytečným chybám.* | **Ukázkový zdrojový kód obsahuje testy** | * ***ano (code coverage 60%+)***
 | **2 body** |
| * ***obsahuje pouze ukázku testů***
 | **1 bod** |
| * ***ne***
 | **0 bodů** |

1. Angličtina je jako nejvhodnější jazyk určena zadavatelem z toho důvodu, že se jedná o nejčastější, dominantní a běžně využívaný programovací jazyk ve vývoji prakticky veškerých softwarových produktů. Z tohoto důvodu je angličtina také užitně nejvýhodnější, nejpoužitelnější a nejpraktičtější programovací jazyk, neboť zajišťuje nejjednodušší následnou kontrolu funkčnosti a provozuschopnosti softwarového produktu a jeho kompatibilitu a napojení na případné další softwarové produkty. [↑](#footnote-ref-1)
2. pozn. Odkaz na uvedený obchodní název zadavatel uvádí pouze pro přesnější a srozumitelnější popis technického požadavku, přičemž připouští i možnost jiného rovnocenného řešení dle § 89 odst. 6 ZZVZ [↑](#footnote-ref-2)
3. pozn. Odkaz na uvedený obchodní název zadavatel uvádí pouze pro přesnější a srozumitelnější popis technického požadavku, přičemž připouští i možnost jiného rovnocenného řešení dle § 89 odst. 6 ZZVZ [↑](#footnote-ref-3)
4. pozn. Odkaz na uvedený obchodní název zadavatel uvádí pouze pro přesnější a srozumitelnější popis technického požadavku, přičemž připouští i možnost jiného rovnocenného řešení dle § 89 odst. 6 ZZVZ [↑](#footnote-ref-4)
5. pozn. Odkaz na uvedený obchodní název zadavatel uvádí pouze pro přesnější a srozumitelnější popis technického požadavku, přičemž připouští i možnost jiného rovnocenného řešení dle § 89 odst. 6 ZZVZ [↑](#footnote-ref-5)