

## 1. Výběr zadání

Vytvořte tříčlenné (preferované), případně dvoučlenné týmy. Vedoucí týmu se přihlásí na jednu z šesti variant. Nejpozději do 31. 10. 2021 se členové týmu ujistí, že spolupráce funguje, nejlépe tím, že začnou na projektu pracovat. Do tohoto termínu lze měnit týmy a zadání. Po tomto datu již změny nejsou možné - projekt by se již pravděpodobně nestihl dokončit. Ostatní problémy v týmu budou řešeny individuálně.

## 2. Cíle projektu

Cílem projektu je navrhnout a implementovat informační systém s webovým rozhraním pro zvolené zadání jedné z variant. Postup řešení by měl být následující:

1. Analýza a návrh informačního systému (analýza požadavků, tvorba diagramu případů užití, modelu relační databáze)
2. Volba implementačního prostředí - databázového serveru a aplikační platformy
3. Implementace navrženého databázového schématu ve zvoleném DB systému
4. Návrh webového uživatelského rozhraní aplikace
5. Implementace vlastní aplikace

## 3. Rozsah implementace

Implementovaný systém by měl být prakticky použitelný pro účel daný zadáním. Mimo jiné to znamená:

- Musí umožňovat vložení odpovídajících vstupů.
- Musí poskytovat výstupy ve formě, která je v dané oblasti využitelná. Tedy nezobrazovat obsah tabulek databáze, ale prezentovat uložená data tak, aby byla pro danou roli uživatele a danou činnost užitečná (např. spojit data z více tabulek, je-li to vhodné, poskytnout odkazy na související data, apod).
- Uživatelské rozhraní musí umožňovat snadno realizovat operace pro každou roli vyplývající z diagramu případů použití (use-case). Je-li cílem např. prodej zboží, musí systém implementovat odpovídající operaci, aby uživatel nemusel při každém prodeji ručně upravovat počty zboží na skladě, pamatovat si identifikátory položek a přepisovat je do objednávky a podobně.

Kromě vlastní funkcionality musí být implementovány následující funkce:

- Správa uživatelů a jejich rolí (podle povahy aplikace, např. obchodník, zákazník, administrátor). Tím se rozumí přidávání nových uživatelů u jednotlivých rolí, stejně tak možnost editace a mazání nebo deaktivace účtů. Musí být k dispozici alespoň dvě různé role uživatelů.
- Ošetření všech uživatelských vstupů tak, aby nebylo možno zadat nesmyslná nebo nekonzistentní data.
  - Povinná pole formulářů musí být odlišena od nepovinných.
  - Hodnoty ve formulářích, které nejsou pro fungování aplikace nezbytné, neoznačujte jako povinné (např. adresy, telefonní čísla apod.) Nenuťte uživatele (opravujícího) vyplňovat desítky zbytečných řádků.
  - Při odeslání formuláře s chybou by správně vyplněná pole měla zůstat zachována (uživatel by neměl být nucen vyplňovat vše znovu).
  - Pokud je vyžadován konkrétní formát vstupu (např. datum), měl by být u daného pole naznačen.
  - Pokud to v daném případě dává smysl, pole obsahující datum by měla být předvyplněna aktuálním datem.
  - Nemělo by být vyžadováno zapamatování a zadávání generovaných identifikátorů (cizích klíčů), jako např. ID položky na skladě. To je lépe nahradit výběrem ze seznamu. Výjimku tvoří případy, kdy se zadáním ID simuluje např. čtečka čipových karet v knihovně. V takovém případě prosím ušetřete opravujícímu práci náповědou několika ID, která lze použít pro testování.
  - Žádné zadání nesmí způsobit nekonzistentní stav databáze (např. přiřazení objednávky neexistujícímu uživateli).
- Přihlašování a odhlašování uživatelů přes uživatelské jméno (případně e-mail) a heslo. Automatické odhlášení po určité době nečinnosti.

## 4. Implementační prostředky

#### 4.1 Uživatelské rozhraní (front-end)

- HTML5 + CSS, s využitím JavaScriptu, pokud je to vhodné.
- Je povoleno využití libovolných volně šířených JavaScriptových a CSS frameworků (jQuery, Bootstrap, atd.)
- Případně lze využít i AJAX či pokročilejší klientské frameworky (Angular, React, Vue, apod.), není to ale vyžadováno.

Rozhraní musí být funkční přinejmenším v prohlížečích Chrome a Firefox (uvažujte aktuálně dostupné verze).

#### 4.2 Implementační prostředí (back-end)

- implicitně jazyk PHP (případně volitelně libovolný PHP framework - Nette, Laravel, Symfony, apod.) - na serveru eva máte dostupný prostor pro studentské stránky:
  - stránky zprovozníte dle návodu v [FAQ](#) - bod 13 (**nezapomeňte, že adresář WWW musí obsahovat platný dokument index.html nebo index.php a soubory musí mít správně nastavená přístupová práva**)
  - používejte kódování UTF-8** - nastavte si soubor .htaccess - dle návodu v [FAQ](#) - bod 14
  - problém s nutností použití vyšší verze php než 5.3 (např. kvůli zvolenému frameworku) řešte dle návodu v [FAQ](#) - bod 15, případně zvolte nějaký vhodný hosting (např. Endora)
- alternativně můžete použít jinou serverovou technologii (např. Python, Javascript/Node.js, Java, C#, Go, Ruby, apod.) a vhodný framework (Django, Flask, Express, Spring, ASP.NET, apod.):
  - podmínkou #1 je, aby byl informační systém dostupný - ověřte si předem, že máte k dispozici vhodný hosting/cloud, na kterém bude schopni IS zprovoznit (např. Heroku, Google Cloud, RedHat Openshift, MS Azure, apod.)
  - podmínkou #2 je, abyste **nepoužili hotový redakční systém, administrační stránky umožňující spravovat obsah** apod. (např. při použití Frameworku Django se žádný uživatel nebude přihlašovat do administrační stránky)
  - v případě použití exotických platforem, které nebudete schopni zprovoznit jinak než lokálně, je nutné se předem domluvit do **31. 10. 2021** (viz kontakt níže) - bude povoleno ve výjimečných případech

#### 4.3 Databázový systém

- implicitně MySQL: lze využít server eva (ve WISu si vytvořte uživatele: Ostatní -> Hesla)
- alternativně můžete použít jiný **relační** databázový systém - PostgreSQL, apod. (který bude dostupný na Vámi zvoleném serveru)
- lze použít dostupné knihovny pro ORM
- použití SQLite není povoleno
- použití nerelační databáze (MongoDB apod.) není povoleno (při případně po domluvě u vlastních zadání)

### **5. Dokumentace**

Součástí projektu je stručná dokumentace k implementaci, která popisuje, které PHP skripty (případně kontrolery, presentery apod. podle zvoleného frameworku) implementují jednotlivé případy použití. Tato dokumentace je součástí dokumentu doc.html, viz níže.

### **6. Odevzdání**

Odevzdání probíhá přes IS FIT. Od okamžiku odevzdání nejméně do **31. 12. 2021** musí být dále funkční aplikace přístupná přes síť Internet na některém fakultním nebo jiném serveru. Tato aplikace musí umožňovat přihlášení pod všemi rolemi uživatelů a musí být naplněna ukázkovými daty, na kterých lze vyzkoušet všechny funkce. Pokud je to technicky možné, použijte umístění <http://www.stud.fit.vutbr.cz/~xlogin00/IIS>. V opačném případě zde nahrajte pouze dokumentaci, v které bude odkaz směřující na Vámi zvolený hosting.

Přes IS se odevzdává jeden archiv pojmenovaný vas\_login.zip obsahující:

- Všechny zdrojové kódy a datové soubory aplikace. Vzhledem k limitu velikosti odevzdaného souboru ve WISu (2 MB) odevzdávejte pouze vlastní kód a data. Neodevzdávejte prosím kódy použitých knihoven a frameworků třetích stran. Místo toho uveďte pouze jejich verze v dokumentaci.
- SQL skript nebo jiný prostředek pro vytvoření a inicializaci schématu databáze.
- Soubor [doc.html](#) obsahující dokumentaci. Stáhněte si [šablonu dokumentace](#), editujte a přiložte k odevzdanému projektu. Respektujte prosím pokyny obsažené v tomto souboru. Některé pokyny mají formu komentářů v HTML kódu šablony.

Za každý tým odevzdává pouze **vedoucí týmu**.

Termín pro odevzdání do IS FIT je **29. 11. 2021**. Po tomto termínu již nelze projekt akceptovat.

## 7. Body

Za projekt je možno získat až 30 bodů.