Strukturovaný posudek oponenta bakalářské práce

Autor/autorka práce: **Jiří Martínek**

Název práce: **Webová aplikace Fantasy 1. hokejová liga**

Obsah práce:

~~Vynikající logická struktura, nadprůměrný obsah i rozsah~~;
Velmi dobrá logická struktura, odpovídající obsah i rozsah;
~~Vyhovující logická struktura, obsah i rozsah~~;
~~Nevyhovující~~

Komentář:

Kvalita řešení a dosažených výsledků:

~~Vynikající~~; ~~Velmi dobrá~~; Vyhovující; ~~Nevyhovující~~

Komentář: Vlastní kód aplikace je na velmi špatné technologické úrovni a obsahuje množství nedostatků.

Formální úroveň:

Vynikající; ~~Velmi dobrá~~; ~~Vyhovující~~; ~~Nevyhovující~~

Komentář: práce je po formální stránce v pořádku a nemám k ní výhrady.

Práce s literaturou:

~~Vynikající~~; Velmi dobrá; ~~Vyhovující~~; ~~Nevyhovující~~

Komentář:

Splnění zadání:

~~Splněno bez výhrad~~; ~~Splněno s menšími výhradami~~; Splněno s většími výhradami; ~~Nesplněno~~

Komentář: textová část práce týkající se zabezpečení aplikace je zcela nedostatečná a nevyhovující

Doplňující informace k práci:

Text práce je v pořádku a nemám k němu zásadnější připomínky. Webová aplikace je funkční a splňuje všechny požadavky. Technická realizace je velmi špatná. Největší chyby dělím do oblastí dle použité technologie:

- HTML - student používá XHTML Strict místo vhodnějšího HTML5. Hlavním problémem však je, že ani úvodní stránku webu, která neobsahuje téměř žádný text, nemá validní. Chyby jsou dokonce i v jednořádkovém importu JavaScriptu.
- CSS - student sice správně základní css pravidla ukládá do samostatného css souboru, ale v některých případech si nedělá žádné starosti ani z přímého vložení značného množství css pravidel přímo do html kódu.
- JavaScript - některé skripty jsou poměrně rozsáhlé, avšak množství opakujícího se kódu je velmi značné a v jednom případě by daný kód šlo zkrátit o cca 90%.
- PHP – celý Php kód je na velmi špatné úrovni. Student nepoužívá ani základní techniky tvorby webových aplikací v Php. Není zde ani náznak MVC, neboť jednotlivé stránky aplikace jsou realizované s pomocí individuálních Php souborů, které přímo obsahují celý HTML kód stránky včetně Php. Není zde ani náznak šablonovacího systému.
- Práce s databází z Php – absolutně nevyhovující. V dnešní době se používá PDO nebo mysqli. Student používá funkci *mysql_query*, které je od Php 5.5 označena jako deprecated. Navíc neřeší sql injection.

- Zabezpečení – aplikace není vůbec zabezpečena proti běžným typům útoků. Bez problémů lze provést i učebnicovou verzi sql injection, kterou použitý Php kód bez problémů provede:
  $player_id = $_GET['id'];
  $result = mysql_query("SELECT player_name FROM 1_czech_league_players WHERE id = $player_id");
- Zabezpečení úpravy sestavy ze strany 29 je v daném php souboru realizováno pouze s využitím JavaScriptu. Pokud v prohlížeči JavaScript jednoduše zakážu, tak jsem uvedené zabezpečení „prolomil".

Dotazy k práci:

1) Co to je SQL injection a jak se proti němu můžeme bránit? Jak byste zabezpečil Vaší aplikaci, aby tento typ útoku nebylo možné provést?
2) Proč jste používal XHTML Strict a nikoliv volnější Transitional, když Váš HTML kód stejně není validní?

Navrhuji hodnocení známkou **dobře** a práci doporučuji k obhajobě.

V Plzni 14.5.2015

Ing. Martin Dostal, Ph.D.