

# Podmienky pre odovzdanie semestrálnej práce z predmetu VAII

---

Jednou z možností ako získať potrebný počet bodov z predmetu *Vývoj aplikácií pre internet a intranet* (VAII) je vypracovanie semestrálnej práce. Podmienky, ktoré práca musí spĺňať, sú uvedené v nasledujúcom texte.

## Termíny

1. Termín odovzdania práce je 12. týždeň semestra počas cvičení za **40 bodov**.
2. Náhradný termín bude cez skúškové obdobie, na tomto termíne je možné za semestrálnu prácu získať max. **30 bodov** a žiadne dodatočné body.

## Práca

1. Každý študent vypracováva prácu **samostatne** a **osobne** ju aj odovzdáva, pričom nie je možné zaslať prácu elektronicky alebo ju odovzdať prostredníctvom iného študenta.
2. **Nesamostatné vypracovanie jednotlivých častí je dôvodom na neabsolvovanie predmetu.**
3. Jednu semestrálnu prácu môžu vypracovať aj viacerí študenti spoločne. Zložitosť takejto práce musí byť adekvátna počtu študentov, pričom každý študent pracuje na svojej časti. Tieto jednotlivé časti musia byť jednoznačne stanovené.
  - a. Pred odovzdaním takejto práce je **potrebné** prebrať tému práce a jej rozdelenie s vyučujúcim, aby ste predišli jej neuznaniu, resp. neuznaniu jednotlivých častí.
  - b. Pokiaľ je nejaká časť hodnotená ako slabá alebo nedostačujúca, bude za ňu strhnutý príslušný počet bodov.
4. Semestrálna práca musí byť **internetová aplikácia**, ktorá využíva nejaké dynamické technológie na strane servera.
5. Prácu je možné prezentovať na vlastnom počítači alebo notebooku, prípadne umiestniť na verejne dostupný webhosting. V prípade potreby je k dispozícii aj školský server (web server + DB server).
6. Za prácu, ktorá je nad rámec popísaných požiadaviek, je možné získať **bonusové body** (napr. o použitie iných frameworkov alebo technológií, aké sú preberané na prednáškach).
7. Dokumentácia k práci je dobrovoľná.

8. Práca sa obhajuje **priamo** pri odovzdávaní, kde študent dostane niekoľko otázok týkajúcich sa riešenia semestrálnej práce. Študent však musí mať prístup ku **zdrojovým kódom** aplikácie počas obhajoby práce, aby bolo možné overiť samostatnosť vypracovania.

## Aplikácia

Výsledná internetová aplikácia je rozdelená na **klientsku a serverovú** časť, ktoré musia spĺňať nasledovné požiadavky:

### Všeobecné požiadavky

1. Zdrojový kód musí byť správne štruktúrovaný, členený a prehľadne sformátovaný. Dôvodom na vrátenie práce je neprehľadný, zle členený alebo často sa opakujúci kód.
2. Pri vytváraní sa snažte dodržiavať postupy *OOP* a *MVC* (prípadne iné architektúry ako *MVVM*, *MVP*, ...), môžete za to dostať **pridané bonusové body**.
3. Hodnotí sa kód, ktorý je napísaný študentom. Pokiaľ je v aplikácii použitý framework, CMS alebo iné hotové riešenie bude hodnotená miera modifikácie alebo implementácie, ktorú študent vytvoril. V odovzdávanej aplikácii však musí byť v dostatočnej miere obsiahnutá aj **vlastná tvorivá činnosť** študenta. Ak chcete odovzdať semestrálnu prácu tohto typu, je vhodné svoj zámer prekonzultovať s vyučujúcim.
4. Aplikácia musí dodržať tieto zásady bezpečnosti:
  - a. Jednoduchá **kontrola vstupu** (na strane servera aj klienta)
  - b. Aplikácia musí obsahovať **časť prístupnú až po prihlásení používateľa** (t.j. musí obsahovať "prihlásenie do webu").
  - c. Dáta o prihlásení sú na strane servera.
  - d. **Heslá užívateľov musia byť zabezpečené** uložené, buď na úrovni databázy alebo aplikačnej logiky vid'. problematika „*password salted hash*“.
5. Klientska strana musí so stranou servera komunikovať **asynchrónne (AJAX)**, min. dvoma spôsobmi z nasledujúceho zoznamu:
  - a. odosielanie formulárov
  - b. editovanie záznamu priamo v tabuľke (*in-place editing*)
  - c. prihlasovanie do aplikácie
  - d. filtrovanie záznamov tabuľky
  - e. načítavanie obsahu tabuľky
  - f. stránkovanie obsahu

g. iné, podľa vlastného výberu

### Server - databáza

1. Musí obsahovať **minimálne 3 zmysluplné entity (databázové tabuľky)**.
2. Databáza **by mala spĺňať normalizáciu**; t.j. žiadne duplicity dát a nezmyselné tabuľky, ktoré sú vytvorené do počtu. Príklady nevhodného použitia napr. :
  - a. *Pre každého užívateľa sa generuje vlastná tabuľka – stačí jedna tabuľka, užívateľov rozdeľuje ich ID atribút*
  - b. *Každá rola užívateľa ma vlastnú tabuľku – do tabuľky užívateľov stačí pridať stĺpec, ktorý nesie informáciu o roliach*

### Server – aplikačná logika

1. Aplikácia vykonáva nad DB všetky operácie – **vkładanie, čítanie, mazanie a upravovanie dát**.
2. Aplikácia využíva všetky entity v DB. Nepoužívaná entita sa neuznáva do počtu.

### Klient - aplikačná logika + GUI

1. Aplikácia musí obsahovať **minimálne 5 podstránok s dynamickým obsahom**.
2. Aplikácia musí mať **responzívny dizajn**, t.j. jej zobrazenie sa musí vhodne prispôbiť rozlíšeniu displeja, na ktorom bude zobrazovaná.
3. Vzhľad a formát aplikácie musí definovať minimálne **20 CSS pravidiel**.
4. Je potrebné vypracovať vlastný klientsky skript o veľkosti **minimálne 25 riadkov**. Skript musí byť zmysluplný a používaný v aplikácii.
5. GUI musí spĺňať pokročilé funkcie pre správu dát v aplikácií. **Zle riešenie** je napr. mazanie záznamu pri **manuálnom zadaní ID** riadku.
6. **Aplikácia musí generovať výstupy podľa W3C štandardov** – resp. študent musí zdôvodniť, prečo aplikácia nezodpovedá štandardom.