

Základné pokyny:

Projekty sa budú robiť po štyroch, pričom sa počíta s tým, že úlohy si medzi sebou rozdelíte rovnomerne.

Zadanie je potrebné odovzdať spakované do prostredia G-Suite najneskôr do 15.5.2020 (23:55) jedným členom tímu. Neskoršie odovzdanie projektu bude penalizované 1 bodom **za každý deň** omeškania na každého člena tímu.

Pre účely ukončenia predmetu je potrebné mať celý projekt umiestnený na školskom serveri. Odovzdanie sa robí cez G-Suite a je tam potrebné vložiť:

- spakované súbory vrátane konfiguračného súboru, v ktorom je potrebné definovať všetky nastavenia,
- sql súbor pre naplnenie databázy,
- adresu umiestnenia, aby sme vedeli, pod koho menom máme projekt hľadať,
- login a heslo pre administrátorský prístup do databázy,
- adresu projektu vo verzionovacom systéme.

Súčasťou projektu bude podstránka s technickou dokumentáciou k projektu a rozdelením úloh medzi jednotlivých členov tímu.

Zameranie projektu

Hlavnou úlohou projektu bude spojazdniť cez API niektorý z voľne dostupných CAS (Computer Aided System), ktorý si je potrebné nainštalovať na niektorý zo serverov členov tímu. Odporúčaný je voľne dostupný softvér Octave, ale v prípade záujmu to môže byť aj iná alternatíva (Maxima, Python riešenia, atď).

V prípade, že použijete nejaké API, ktoré nájdete na Internete a ktoré si prispôbíte pre svoje účely, treba túto skutočnosť uviesť v technickej dokumentácii, inak sa to bude považovať za plagiátorstvo. Ja o takomto API neviem, ale je veľká pravdepodobnosť, že niečo existuje.

Funkčnosť Octave príkazov si je možné na začiatku vyskúšať bez inštalácie tohoto prostredia na stránke: <https://octave-online.net/>

Nezabudnite na to, že sa hodnotí aj grafický dizajn vytvorenej aplikácie, vhodne navrhnuté členenie, ľahkosť orientácie v prostredí.

Pamätať by ste mali aj na zabezpečenie celej aplikácie.

Na vypracovanie projektu je možné použiť už aj PHP framework.

Požiadavky na projekt

1. Pri práci na projekte je potrebné používať verzionovací systém, napr. „github“, „gitlab“, „bitbucket“.
2. Vytvorená webstránka bude navrhnutá ako dvojazyčná (slovenčina, angličtina).
3. API k nainštalovanému CAS je potrebné vytvoriť v takom rozsahu, aby ste pomocou neho dokázali vyriešiť individuálne úlohy v rámci projektu, ktoré budú zamerané na animáciu objektu

a vykresľovanie grafu. V prípade, že výpočet hodnôt bude pre tieto činnosti príliš rýchly, spomaľte ho **na strane servera**. Prípadný koeficient spomalenia definujte medzi ostatnými hodnotami v konfiguračnom súbore.

4. Jednotlivé výstupy z CAS budú poskytované užívateľovi na základe API kľúča, ktorý bude zadaný v konfiguračnom súbore a bez ktorého zadania nebude možné poskytovanú službu používať.
5. Požiadavky na výpočet budú do CAS zasielané dvoma spôsobmi:
 - z individuálnych úloh,
 - z formulára na web stránke (1x textarea na zadanie príkazu, 1x tlačidlo na odoslanie požiadavky, 1x output na výpis výstupu). Pri formulári stačí realizovať iba zadané príkazy a nie je potrebné si pamätať žiadne premenné. Stále platí, že stačí riešiť rozsah príkazov definovaných v bode 3 s tým, že v rámci tohto bodu nemusíte riešiť vykresľovanie grafov. Predpokladám, že príkaz „1+1“ sa odskúšať bude dať.
6. Všetky požiadavky zasielané do CAS je potrebné logovať, pričom je potrebné uchovávať
 - dátum a čas vykonania požiadavky,
 - odoslané príkazy,
 - info, či príkazy boli vykonané korektne alebo sa tam vyskytla nejaká chyba,
 - prípadné info o chybe.
7. Požiadavky z bodu 6 je potrebné vedieť exportovať do CSV a PDF súboru.
8. Na stránke je treba urobiť aj jednoduchý popis vytvoreného API. Tento popis je tiež potrebné umožniť vygenerovať do PDF súboru.
9. Na stránke robte aj štatistiku o tom, ktorá z individuálnych úloh bola koľkokrát využívaná. Okrem zobrazenia tejto informácie na stránke, umožnite jej odoslanie aj na užívateľom zadaný mail.

Individuálne úlohy:

Úlohou bude vytvoriť animáciu dynamického systému a priebežne generovaný graf sledovaných výstupov na základe užívateľom definovaných parametrov.

V rámci individuálnej úlohy bude potrebné animovať správanie sa jedného zo zadaných štyroch dynamických systémov:

- inverzné kyvadlo
(<http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>),
- guľička na tyči
(<http://ctms.engin.umich.edu/CTMS/index.php?example=Suspension§ion=SystemModeling>),
- tlmič automobil
(<http://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam§ion=SystemModeling>),

- náklon lietadla
(<http://ctms.engin.umich.edu/CTMS/index.php?example=AircraftPitch§ion=SystemModeling>).

Každý v tíme si zvolí jeden z týchto modelov, t.j. modely si medzi sebou rozdelíte.

Hyperlinky v zátvorkách smerujú na stránky, kde je vysvetlená fyzikálna podstata týchto systémov, v prípade, že by to niekoho zaujímalo. Na naprogramovanie úlohy však vystačíte s kódmi, ktoré sú uvedené v osobitnom súbore modely.pdf a príkazy si môžete nakopírovať aj z priložených textových súborov, kde je uvedená aj presná linka stránky, odkiaľ boli tieto príkazy čerpané. To znamená, že na naprogramovanie projektu nie je vôbec potrebné rozumieť matematickej a fyzikálnej podstate problému.

1. Animácia sa robí na základe hodnôt, ktoré vygeneruje vaše API.
2. Riešenie úlohy bude graficky zapracované do vytvorenej web stránky.
3. Animáciu riešte buď pomocou canvasu alebo svg. Môže ísť o 2D alebo 3D animáciu. Na vykreslenie animácie môžete využiť niektorú z javascriptových grafických knižníc, ktoré sú voľne dostupné na Internete. Napríklad fabric.js, p5.js, atď.
Na internete nájdete rôzne grafické prevedenia jednotlivých modelov, stačí si len vybrať, ktoré vám najviac vyhovuje. Vyhľadávať môžete obrázky pomocou nasledujúcich kľúčových slov: inverted pendulum, beam and ball, suspension system car – stačí však pekné vyhotovenie vozidla a kolesa, aircraft pitch control – dôležité je hlavne telo lietadla a zadná klapka)
4. Na vykreslenie grafu je možné použiť napr. Plotly, Flot, jqPlot, Highcharts, D3.js, ... alebo si zvoliť niečo podľa vlastných preferencií.
5. Užívateľ bude zadávať iba 1 vstup a sledovať 2 výstupy, ktoré sa využijú pri animácii a vykresľovaní grafov:
 - *pri kyvadle* – vstup: požadovaná nová poloha kyvadla r , výstup: aktuálna pozícia kyvadla $x(:,1)$ a aktuálny uhol kyvadla (náklon vertikálnej tyče – uhol v radiánoch) $x(:,3)$,
 - *pri guľičke na tyči* – vstup: požadovaná nová poloha guľičky na tyči r , výstup: aktuálna pozícia guľičky $N \cdot x(:,1)$ a aktuálny náklon tyče (uhol v radiánoch) $x(:,3)$,
 - *tlmič auta* – vstup: požadovaná výška skokovej prekážky r , výstup: aktuálna pozícia vozidla $x(:,1)$ a aktuálna pozícia kolesa $x(:,3)$,
 - *náklon lietadla* – vstup: požadovaný nový náklon lietadla r , výstup: aktuálny náklon lietadla $x(:,3)$ a aktuálny náklon zadnej klapky $r \cdot \text{ones}(\text{size}(t)) \cdot N - x \cdot K'$.

Treba však kontrolovať, či je zmyslupne zadávaný a ak nie, tak upozorniť na to pomocou tooltipu.
6. Na stránke budú 2 checkboxy, kde si užívateľ zvolí, či chce sledovať animáciu, graf alebo obidve veci súčasne.
7. **Nepovinný bonus za 5 bodov:** Na stránke si bude môcť užívateľ zadať svoje meno. V prípade jeho zadania bude toto meno k dispozícii všetkým ostatným užívateľom, ktorí sú v danom okamihu pripojení na stránku. Ak sa rozhodnú, tak sa budú môcť pripojiť na stránku daného zvoleného užívateľa a pasívne sledovať ako robí experimentovanie.