

# Пројектни задатак из Пројектовања информационих система

10. новембар 2023.

Креирати веб апликацију која представља портал електронских новина. У наставку следи опис апликације, као и листа функционалности и захтева које апликација треба да задовољи.

## 1 Функционални захтеви

1. корисници на порталу могу да прочитају вести (кориснику се прво приказују актуелне вести, али свакако му је доступна и архива коју може да претражи)
2. вести су дефинисане насловом, текстом (текст може да садржи и слике и видео линкове, може да се форматира као подебљан и курзив текст, може да садржи наслове и поднасловe) и таговима (ради лакшег индексирања и претраге). Сваки чланак припада тачно једној рубрици (политика, црна хроника, свет, спорт, забава, култура,...).
3. вест је могуће претражити по наслову, таговима и/или датуму
4. корисник може да *лајкује* вест, односно означити да му се вест свиђа или не свиђа. Такође, корисник може да коментарише вести. За ове акције, није потребна никаква пријава, већ корисник може да унесе једнократно корисничко име при уносу коментара. Коментаре је такође могуће означити као *свиђа ми се* и *не свиђа ми се*. Уз вести и коментаре приказује се и број свиђања и не-свиђања.
5. да би неко из редакције написао нову вест, потребно је да буде регистрован на CMS овог сајта. Разликујемо две роле:
  - **новинач** - новинар може да креира нову вест у оквиру неке рубрике. Једном када је вест написана, она се чува у драфт стању као радна верзија. Докле год је у овом стању, новинар може да је измени или обрише. Када је завршио са писањем те вести, новинар је шаље уреднику на прегледање. Докле год је вест у том стању, вест је и даље подложна изменама и брисању. Оног тренутка када уредник одобри ту вест, новинар више не може да је самостално обрише или измени. Једном када је вест одобрена, свака њена измена захтева ново одобрење. Да би вест која је одобрена била обрисана, тај захтев такође мора бити одобрен од стране уредника. Једном када је вест одобрена, она постаје видљива на сајту

(радна верзија није видљива на сајту). Систем привилегија је дизајниран тако да новинари могу да објављују само вести у оквиру својих рубрика - дакле, једном када је новинар додат у систем, додељују му се и рубрике. Тако, новинар може да додаје чланке само у оне рубрике које су му додељене.

- **уредника** - уредник може бити надлежан за једну или више рубрика, или с друге стране, може бити главни уредник. Уредник рубрике може да одобрава и брише само чланке у оквиру те рубрике, док главни уредник има уредничке привилегије над свим рубрикама.
- Подразумевати да је на почетку у систем додат само главни уредник. Само главни уредник може да региструје остале новинаре. Уредник може да додели своју рубрику неком новинару или да је уклони. Главни уредник има сва уредничка права, а додатно, може да прогласи новинара уредником или да га уклони с тог задужења.

6. подразумевати да је потребно имплементирати систем пријаве у систему

## 2 Технички и нефункционални захтеви

- сав програмски код треба да буде похрањен у приватном репозиторијуму на платформи за верзионисање пројеката *GitHub*. Апликацију изградити у неком од следећих језика: Python, Java, C, C++, PHP или JavaScript. За израду графичког интерфејса (сучеља), дозвољено је користити било које библиотеке.
- новински чланци треба да буду трајно сачувани, односно, потребно их је складиштити у бази података. Студентима се оставља на избор који DBMS ће користити.
- компоненте је потребно контејнеризовати (сервисне компоненте и базу података)
- \* потребно је аутоматизовати ток извршења континуалне интеграције (CI)
- потребно је покрити бар део кода тестовима - *unit* и интеграционим, и то тако да:
  - *unit* тестовима покрити случајеве одобрења новинске објаве и операције над њима у разним стањима - драфт, послат на прегледање (одобрење) и објављен. Под операције над објављеним чланком укључити и коментарисање и *like/dislike* функционалност коју има читалац.
  - \* интеграционим тестовима покрити случајеве претраге чланака - по наслову, рубрици, таговима и датуму
- \* узећемо у обзир да имамо доступну архиву од претходних 5 година која је сачувана и коју желимо да учитате у вашу базу података. Водити се следећом рачуницом: нека сваког дана новине објаве 5 чланака у разним рубрикама (можете узети насумичну расподелу, више објава може бити из исте рубрике). Према томе  $5 * 365 * 5 = 9125$  новинских чланака. Потребно је да се при покретању система, покрене иницијализација података - написати скрипту која ће генерисати тражени број објава. Подразумевати да неке објаве треба да садрже и слике и видео линкове. Текст, слике и видео линкови тих објава се остављају вама на избор.

### 3 Предаја пројеката и оцењивање

- пројекте је могуће радити самостално или у оквиру четворочланих тимова. Тимови треба да имају следеће роле:
  - *backend engineer* - задужен за развој бизнис логике и моделирање података
  - *frontend engineer* - задужен за развој корисничког интерфејса
  - *quality assurance engineer* - задужен за писање тестова. Писање *unit* тестова је добра (обавезна) пракса за софтверске инжењере и у том смислу неко ко развија неки код на *backend-у* би требало да пише и те тестове. Ипак, ради балансирања оптерећења, ову ставку ће имплементирати члан тима који има ролу QA инжењера. Према томе, овај члан тима ће бити задужен за писање тестова (и *unit* и интеграционих).
  - *DevOps/system/infra engineer* - задужен за подешавање и одржавање инфраструктуре на којој апликација ради. Овај члан тима ће бити задужен за контејнеризацију сервиса, писање CI и додатно, за писање скрипте која ће иницијализовати податке у оквиру система.
- студенти који буду самостално радили пројекте нису у обавези да одраде ставке означене звездицом.
- сваки студент је у обавези да направи **бар 5 смислених *commit-ова***. За студенте који не буду испунили овај захтев, сматраће се да нису активно учествовали у изради пројекта и њихово ангажовање неће бити узето у обзир.
- 
- предаја пројекта је могућа у три рока. Након ових термина, неће бити могуће предати пројекте и сматра се да студент/тим није испунио захтеве пројектног задатка, чиме добија 0 бодова и губи право полагања испита у текућој школској години. Да би обавеза израде пројектног задатка била испуњена, потребно је освојити минимално **1 поен**. Термини предаје пројеката:
  - **до 15. фебруара 2024. године (закључно са 14. фебруаром)** - могуће је освојити максималних 40 поена.
  - **до 1. марта 2024. године (закључно са 29. фебруаром)** - могуће је освојити 30 поена.
  - **до 15. марта 2024. године (закључно са 14. мартom)** - могуће је освојити 20 поена.
- сваки пројекат, поред програмског кода, треба да садржи и пратеће упутство у форми README фајла који ће бити део пројекта на *GitHub* репозиторијуму. У оквиру овог фајла ће потребно укратко описати архитектуру саме апликације, упутство за покретање апликације, навести побољшања, као и навести чланове тима. Студенти који тимски раде пројекат не морају да предају никакву додатну документацију. Студенти који самостално израђују пројекат су у обавези да напишу и додатну документацију коју је дефинисао професор Ђорђевић, а која је наведен на страници овог предмета на *Moodle-у* порталу.

- завршетак израде пројекта, студент(и) потврђује слањем *email*-а професорима Ђорђевићу и Пантелићу **искључиво на следеће адресе:** adjordjevic@kg.ac.rs и npantelic.fin@gmail.com или npantelic@kg.ac.rs
- одбрана пројекта се састоји од презентације урађеног пројекта у виду демонстрације, питања и потенцијалних захтева за изменама, тј. додавања нових или измене постојећих функционалних и/или нефункционалних захтева.
- за студенте који тимски раде пројекат важи правило 60/40 - 60% од укупног броја поена носи ваш тимски учинак, а 40% ваш лични учинак.
- подразумевати да студенте које самостално раде пројекат нагледа проф. Ђорђевић, док тимове нагледа предметни сарадник Пантелић.