

Пројектни задатак из Пројектовања информационих система и база података

6. октобар 2024.

Креирати веб апликацију за креирање и управљање формуларима, попут *Google forms* сервиса. У наставку следи опис апликације, као и листа функционалности и захтева које апликација треба да задовољи.

1 Функционални захтеви

1. да би се апликација користила, потребно је имати налог на платформи. Стога је потребно имплементирати систем регистрације корисника.
2. само пријављени корисници могу да креирају форму. Форму могу попуњавати пријављени корисници, док је неке формуларе могуће попунити и без пријаве, зависно од тога да ли је то дозвољено у подешавању самог формулара.
3. пријављени корисници креирају формуларе задајући притом:
 - (а) име формулара
 - (б) опис формулара
 - (в) индикатор да ли форму могу попунити и непријављени корисници или је обавезно да буду пријављени при попуњавању исте
 - (г) листу питања (где се свако питање дефинише као обавезно или необавезно). Свако питање има свој тип одговора, и то:
 - кратак текстуални одговор - до 512 карактера
 - дуг текстуални одговор - до 4096 карактера
 - више понуђених одговора, само један се бира
 - више понуђених одговора, бира се више одговора (могуће је задати број захтеваних одговора)
 - један нумерички одговор. Задаје се листа бројева, или цела листа бројева или скала бројева са кораком. Нпр. захтев -4 до 20 уз корак 3 би генерисао следећу листу бројева: -4, -1, 2, 5, 8, 11, 14, 17, 20
 - датум
 - време

4. код сваког питања је могуће додати слику у питање или одговор (начин и стил приказа слика у оквиру питања и одговора се оставља вама на избор)
5. свако питање је могуће клонирати
6. свако питање је могуће изменити
7. свако питање је могуће обрисати
8. могуће је променити редослед питања
9. корисник може да подели линк за попуњавање форме
10. корисник може да дода колабораторе на формулар у улози посматрача или уредника, као и да уклони исте. Сви уредници, као и креатор могу да измене све у формулару (измене назив, опис, питања, клонирају и обришу иста), али и да закључају формулар (није могуће попунити такав формулар). Само власник формулара може да управља колабораторима на том формулару.
11. при пријави на сервис, учитавају се сви формулари код којих је корисник креатор или колаборатор
12. сви колаборатори на формулару могу да прегледају резултате и то:
 - појединачне одговоре
 - групне одговоре (груписане по питању)
 - извезу питања и одговоре у `xlsx` фајл

2 Технички и нефункционални захтеви

- Апликација треба да буде имплементирана у форми микросервиса. Сегрегација сервиса и њихове одговорности се остављају студентским тимовима као задатак.
- Сав програмски код треба да буде похрањен у приватном репозиторијуму (једном или више њих) на платформи за верзионисање пројеката *GitHub*. Апликацију израдити у неком од следећих језика: Python, Java, C#, C++, PHP или JavaScript. За израду графичког интерфејса (сучеља), дозвољено је користити било које оквире и библиотеке.
- подаци о корисницима и формуларима треба да буду трајно сачувани, односно потребно их је складиштити у бази података. Студентима се оставља на избор који систем за управљање базом података ће користити.
- компоненте је потребно контејнеризовати (сервисне компоненте и базе података). Сервиси треба да буду подигнути користећи *Docker compose* мултиконтејнер алат или *Kubernetes* оркестрациони алат.
- потребно је аутоматизовати ток извршења континуалне интеграције (CI)

- потребно је покрити код тестовима - *unit* и интеграционим, и то:
 - *unit* тестовима (пишу чланови тима који имплементирају backend функционалности) покрити случајеве:
 - * регистрацију и CRUD операције над корисницима
 - * CRUD операције над формуларима
 - * преглед попуњених формулара (појединачни и агрегациони резултати)
 - интеграционим
 - * [UI] регистрација корисника
 - * [UI] пријава корисника
 - * [API] CRUD операције над корисницима
 - * [API] CRUD операције над формуларима
 - * [UI] претрага формулара по називу
 - * [API] попуњавање формулара. Дефинисати формулар који садржи све типове питања.
 - * [API] затварање формулара
 - * [API] додавање колаборатора на формулар
 - * [API] преглед попуњених формулара (појединачни и агрегациони резултати)
 - мануелно тестирати попуњавање формулара на UI и генерисање *xlsx* извештаја

3 Предаја пројекта и оцењивање

- пројекти се ради у оквиру петочланих тимова. Сваки тим треба да има име и тим лидера. Тимови треба да имају следеће роле:
 - *backend engineer* - задужен за развој бизнис логике и моделирање података, као и писање *unit* тестова.
 - *frontend engineer* - задужен за развој корисничког интерфејса
 - *quality assurance engineer* - задужен за писање интеграционих и end-to-end тестова
 - *DevOps* - задужен за подешавање и одржавање инфраструктуре на којој апликација ради. Овај члан тима ће бити задужен за контејнеризацију сервиса, писање CI и Kubernetes конфигурација. Подразумевано је да ове активности одузму до 2/3 ефективног времена. Остатак времена, овај члан тима треба да подреди развоју backend или frontend функционалности.
- расподела чланова студената међу ролама је ствар тимске одлуке. Неке опције су 1 backend, 1 frontend, 1 QA, 1 DevOps који помаже при развоју BE или FE, пети члан тима ради
 - само backend

- само frontend
- fullstack

У пријавном формулару, овај члан ће бити третиран као joker члан.

- сваки студент је у обавези да направи **бар 5 смислених *commit*-ова**. За студен-те који не буду испунили овај захтев, сматраће се да нису активно учествовали у изради пројекта и њихово ангажовање неће бити узето у обзир. Комитови треба да буду смислени и направљени с временском разделом, а не последњег дана. Такви ангажмани неће бити узети у обзир.
- водите рачуна о комит порукама. Оне треба да буду смислене и сврсисходне. Примера ради:

- добре комит поруке

```
Implement the authentication logic based on JWT
Update the README file with the team member details
Define data models and accompanying SQL modeling scripts
```

- лоше комит поруке

```
Minor change
Fix the bug
Another test added
```

- Пријава тимова и студената се врши преко формулара доступних на страници пред-мета. Рок за пријаву је 16. октобар до 20:00. Студенти и тимови који се не пријаве биће распоређени у тимове насумично.
- 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 **12 поена**. Термини предаје пројектата:
 - до **15. фебруара 2025. године (закључно са 14. фебруаром)** - могуће је освојити максималних 40 поена.
 - до **1. марта 2025. године (закључно са 28. фебруаром)** - могуће је освојити 30 поена.
 - до **15. марта 2025. године (закључно са 14. мартом)** - 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, ток извршавање функционалности и упутство за покретање апликације, навести побољшања, као и навести чланове тима.

- 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 и питања. Презентација пројектног рада траје тачно 15 минута, након чека се постављају питања. Очекивано је да пројекат представи један члан тима, тим лидер или више студената по избору тима.
- преглед бодовања функционалности:
 1. регистрација корисника - 10%
 2. CRUD корисника - 7.5%
 3. пријава корисника - 10%
 4. претрага формулара - 10%
 5. креирање формулара (са свим типовима питања) - 12.5%
 6. клонирање питања - 5%
 7. ажурирање питања - 7.5%
 8. брисање питања - 5%
 9. закључавање формулара - 5%
 10. менаџмент формуларима - 7.5%
 11. преглед одговора - 10%
 12. извоз у *xlsx* формат - 10%
- преглед бодовања DevOps ставки:
 1. контејнеризација сервиса - 65%
 2. CI сервиса - 35%