**1) Tema: Razvoj Knowledge management veb aplikacije**

**Tehnologije i arhitektura rešenja:**

Arhitektura treba da bude MVC ili neka varijacija MVC paterna. RESTfull arhitektura.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery, JS frejmvork (Ember ili Angular)
* Back-end: Symfony/Laravel
* DB: kombinacija MySQL i MongoDB. Funkcionalnosti vezane za pretragu realizovati nad MongoDb

**Korisnički zahtev:**

Potrebno je razviti aplikaciju za upravljanje znanjem u okviru jednog poslovnog sistema. Osnovni cilj aplikacije je da omogući upravljanje informacijama iz različitih izvora znanja, pretragu i deljenje znanja u okviru poslovnog sistema.

Aplikacija treba da obuhvati sledeću listu funkcionalnosti:

* Omogućiti unos/deljenje "znanja" u različitim oblicima (files, documents, photos, pdfs, emails, web links, videos, itd.) i sa više izvora. Omogućiti podešavanje vidljivosti sadržaja na nivou grupe/odeljenja.
* Omogućiti označavanje sadržaja preko ključnih reči, tagova i sl.
* Omogućiti postavljanje komentara i ocene nekog sadržaja
* Napredna pretraga. User defined filters
* Klasifikacija i kategorizacija sadržaja
* Postavljanje pitanja i odgovori, po uzoru na http://stackoverflow.com/
* Email notifikacije korisnicima ako je ispunjen neki kriterijum. Na primer: kada se postavi novi sadržaj.
* Potrebno je obezbediti postojanje minimum dve uloge u sistemu:

- običan korisnik

- administrator

Potrebno je omogućiti upravljanje grupama korisnika i ubacivanja korisnike u grupe/odeljenja

* Napraviti REST API po uzoru na http://www.redmine.org/projects/redmine/wiki/Rest\_api
* Autentikacija na osnovu interne baze ali i eksternog izvora kao što je LDAP server.

Opciono: Mobilna aplikacija

**Okvirni predlog sadržaja:**

1. Uvod
2. Knowledge management. Analiza aplikacija za knowledge management
3. Opis tehnologija primenjenihu razvoju rešenja
4. Razvoj Knowledge management aplikacije
5. Zaključak
6. Literatura

**2) Tema: Implementacija servisa za analizu cena robe u supermarketima**

**Tehnologije i arhitektura rešenja:**

Arhitektura treba da bude MVC ili neka varijacija MVC paterna. RESTfull arhitektura.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery, **obavezno JS frejmvork** (Ember ili Angular), D3 ili Google Chart API
* Back-end: po izboru, predlog je da bude neki PHP frejmvork
* DB: kombinacija MySQL i MongoDB.

**Korisnički zahtev:**

Potrebno je razviti servis koji omogućava sakupljanje cena svih proizvoda u supermarketima: RODA, IDEA, MAXI, TEMPO, DIS i METRO. Konceptualno, aplikacija može da liči na http://cenoteka.rs/. Aplikacija prikazuje korisnicima informacije o cenama u pomenutim supermarketima. U principu, aplikacija liči na elektronsku prodavnicu, koja omogućava pretragu proizvoda i kreiranje kupovne korpe

Aplikacija treba da obuhvati sledeću listu funkcionalnosti:

* Prikaz kategorija proizvoda
* Napredna pretraga i sortiranje proizvoda
* Prikaz proizvoda sa podacima o cenama za svaki od marketa. Označiti najnižu i najvišu cenu. Uz svaki proizvod treba pokupiti i slike sa sajtova marketa. Proizvodi se mogu dodati u korpu.
* Kreiranje standardne kupovne korpe. Voditi računa o tome da proizvodi u korpi moraju biti iz istog marketa
* Vizuelizacija podataka o cenama za svaki proizvod.
* Omogućiti postavljanje komentara uz proizvod
* Mogućnost exporta, štampanja korpe i sl.
* Potrebno je obezbediti postojanje minimum dve uloge u sistemu:

- običan korisnik

- administrator

* Napraviti REST API po uzoru na http://www.redmine.org/projects/redmine/wiki/Rest\_api
* Podatke treba osvežavati što je češće moguće.

**Mobilna aplikacija: Obavezna**

Uzor: [Google shopper aplikacija](http://www.google.com/shopping). Opis zahteva mobilne aplikacije:

* korisnik može da skenira bar kod proizvoda u nekom marketu ili da unese naziv proizvoda i automatski se učitavaju cene odabranog proizvoda
* napredna pretraga
* prikaz lokacija i najkraćih puteva do izabranog objekta/supermarketa

**Okvirni predlog sadržaja:**

* Uvod
* Razvoj veb robota
* Opis tehnologija primenjenihu razvoju rešenja
* Implementacija servisa za analizu cena robe u supermarketima
* Integracija servisa u okviru proizvoljnog veb sajta razvijenog na Wordpress platformi
* Plan promocije
* Zaključak
* Literatura

**3) Tema: Integracija veb aplikacije sa mehanizmima plaćanja**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery
* Back-end: po izboru
* DB: MySQL

**Korisnički zahtev:**

Potrebno je razviti module/klase za integraciju veb aplikacije sa onlajn sistemima plaćanja: PayPal, QVoucher, Autorize.net, Bitcoin, Skrill i sl. Ukratko, integracija podrazumeva mogućnost da se u okviru proizvoljnog e-commerce rešenja korisnicima može obezbediti plaćanja putem nabrojanih metoda.

Za svaki od metoda plaćanja potrebno je implementirati modul, koji sadrži:

* Deo za podešavanje integracije
* Deo za podešavanje prikaza na front-end-u
* Prikaz na front-end delu

Opciono: Ispitati integraciju sa PayPall wallet ili Google wallet

1. Uvod
2. Opis najznačajnijih mehanizama plaćanja
3. Opis tehnologija primenjenihu razvoju rešenja
4. Opis implementiranih modula
5. Zaključak
6. Literatura

**4) Tema: Platforma za microlearning**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery
* Back-end: Symfony
* DB: MySQL

**Korisnički zahtev:**

Potrebno je napraviti aplikaciju po uzoru na https://coursmos.com/. Aplikacija omogućava upravljanje video kursevima, koji se sastoje iz nekoliko video lekcija.

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Kreiranje kurseva. Svaki kurs se sastoji iz jedne ili više video lekcija, koje se kreiraju postavljanjem video zapisa. Svaki kurs i lekciju treba označiti nazivom, tagovima, ključnim rečima i sl. Svaki kurs treba označiti kao besplatan ili plaća se uz informaciju o ceni.
* Obezbediti funkcionalnosti za povezivanje na društvenim mrežama
* Omogućiti postavljanje komentara i ocene nekog sadržaja
* Napredna pretraga. User defined filters
* Klasifikacija i kategorizacija sadržaja
* Mogućnost slanja zahteva za ubacivanjem određenog kursa (request opcija na https://coursmos.com/ )
* Obezbediti upravljanje pristupom kursevima, u zavisnosti od tipa članstva.
* Email notifikacije korisnicima ako je ispunjen neki kriterijum. Na primer: kada se postavi novi sadržaj.
* Omogućiti autentikaciju na osnovu openID protokola!

Mobilna aplikacija: opciono

**Okvirni predlog sadržaja**

1. Uvod
2. Microlearning
3. Analiza platformi za microlearning
4. Opis tehnologija primenjenihu razvoju rešenja
5. Implementacija microlearning platforme
6. Plan prmocije
7. Zaključak
8. Literatura

**5) Tema: Razvoj Moodle plugina za ocenjivanje resursa i komentarisanje**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. Slediti Moodle standarde u pogledu arhitekture i pisanja koda

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery
* Back-end: PHP
* DB: MySQL

**Korisnički zahtev:**

Moodle plugin koji omogućava da se za svaki pojedinačni resurs (fajl, stranica, folder) ulogovanim korisnicima omogući ostavljanje komentara i ocenjivanje (rating)

Okvirni predlog sadržaja

1. Uvod
2. Razvoj Moodle plugina
3. Analiza postojećih plugina u kontekstu teme
4. Razvoj Moodle plugina za ocenjivanje resursa i komentarisanje
5. Integracija Moodle plugina u okviru moodle.elab.fon.bg.ac.rs
6. Postavljanje plugina u direktorijum Moodle zajednice
7. Zaključak
8. Literatura

**6) Tema: News reader aplikacija**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery, Ember.js/Meteor.js
* Back-end: Symfony/Laravel
* DB: MySQL

**Korisnički zahtev:**

Potrebno je napraviti aplikaciju po uzoru na http://goodnoows.com/. Aplikacija omogućava upravljanje vestima koje dolaze preko određenih strimova. Predstavlja personalnog agregatora novosti koji se učitavaju sa news sajtova i portala

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Ubacivanje RSS feedova kao izvora vesti
* Jedinstveni prikaz vesti na početnoj stranici, po uzoru na http://goodnoows.com/
* Obezbediti funkcionalnosti za povezivanje na društvenim mrežama
* Napredna pretraga. User defined filters
* Klasifikacija i kategorizacija sadržaja
* Raznovrsni načini prikaza vesti
* Email notifikacije korisnicima ako je ispunjen neki kriterijum. Na primer: kada se postavi novi sadržaj.
* Omogućiti autentikaciju na osnovu openID protokola!

Mobilna aplikacija: treba da se napravi analogno sa Feedly aplikacijom - https://feedly.com/i/welcome

Okvirni predlog sadržaja

1. Uvod
2. Analiza newsreader servisa
3. Opis tehnologija primenjenih u razvoju rešenja
4. News reader aplikacija
5. Plan promocije
6. Zaključak
7. Literatura

 **7) Social network analysis aplikacija**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC. FB Graph API

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery, Ember.js/Meteor.js, D3
* Back-end: Symfony/Laravel
* DB: MySQL

**Korisnički zahtev:**

Aplikacija predstavlja personalni servis za analizu podataka vezanih za "prijatelje" korisnika aplikacije, po uzoru na http://www.wolframalpha.com/facebook/. Aplikacija treba da omogući sledeće informacije: odakle su FB prijatelji, koje filmove su gledali, koja mesta su posetili i druge informacije.

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Jedinstveni prikaz podataka na početnoj stranici
* Obezbediti funkcionalnosti za povezivanje na društvenim mrežama
* Napredna pretraga. User defined filters
* Klasifikacija i kategorizacija sadržaja
* Raznovrsni načini prikaza podataka
* Omogućiti autentikaciju na osnovu openID protokola!
* Napraviti REST API po uzoru na http://www.redmine.org/projects/redmine/wiki/Rest\_api

Mobilna aplikacija: opciono

**Okvirni predlog sadržaja**

1. Uvod
2. Analiza alata za analizu društvenih mreža
3. Opis tehnologija primenjenih u razvoju rešenja
4. Social network analysis aplikacija
5. Plan promocije
6. Zaključak
7. Literatura

**8) Servis za pretragu radova**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery.
* Back-end: Symfony/Laravel ili Node.js
* DB: MySQL

**Korisnički zahtev:**

Aplikacija predstavlja online direktorijum svih tipova završnih radova. Ovo podrazumeva mogućnost pretrage radova po različitim kriterijumima.

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Omogućiti označavanje radova preko ključnih reči, tagova i sl.
* Omogućiti postavljanje komentara radova
* Napredna pretraga. User defined filters
* Klasifikacija i kategorizacija sadržaja
* Email notifikacije korisnicima ako je ispunjen neki kriterijum. Na primer: kada se postavi novi sadržaj.
* Potrebno je obezbediti postojanje minimum dve uloge u sistemu:

- običan korisnik

- administrator

* Napraviti REST API po uzoru na http://www.redmine.org/projects/redmine/wiki/Rest\_api

Mobilna aplikacija: Mobilna aplikacija treba da omogući prikaz radova po različitim kriterijumima

**Okvirni predlog sadržaja**

1. Uvod
2. Onlajn biblioteke i servisi za pretragu
3. Opis tehnologija primenjenih u razvoju rešenja
4. Servis za pretragu radova
5. Plan promocije
6. Zaključak
7. Literatura

**9) Aplikacija za administraciju mentorstva, datume i sl.**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura. MVC.

Predloženi stack tehnologija po slojevima:

* Front-end: HTML, Bootstrap, jQuery.D3
* Back-end: Symfony/Laravel ili Node.js
* DB: MySQL

**Korisnički zahtev:**

Aplikacija treba da omogući evidenciju mentorstava nastavnika na završnim radovima, kao i predaju radova od strane studenata.

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Omogućiti označavanje radova preko ključnih reči, tagova i sl.
* Omogućiti postavljanje komentara radova
* Omogućiti upravjanje verzijama rada
* Napredna pretraga. User defined filters
* Dvosmerna integracija sa WP ajax calendar plugina
* Klasifikacija i kategorizacija sadržaja
* Email notifikacije korisnicima ako je ispunjen neki kriterijum. Na primer: kada se dodeli mentorstvo, postavi aplikacija i sl.
* Potrebno je obezbediti postojanje minimum dve uloge u sistemu:

- običan korisnik

- administrator

* Napraviti REST API po uzoru na http://www.redmine.org/projects/redmine/wiki/Rest\_api
* Napraviti servis za izveštavanje. Generisanje izveštaja u raznim formatima, ad hoc izveštaji,

Mobilna aplikacija: Mobilna aplikacija treba da omogući prikaz radova po različitim kriterijumima.

**Okvirni predlog sadržaja**

1. Uvod
2. Opis tehnologija primenjenih u razvoju rešenja
3. Opis aplikacije za administraciju mentorstva, datume i sl.
4. Zaključak
5. Literatura

**10) ELAB Android/iOS aplikacija**

**Tehnologije i arhitektura rešenja:**

RESTfull arhitektura

Predloženi stack tehnologija po slojevima:

* Android/ iOS

**Korisnički zahtev:**

Aplikacija treba da omogući prezentaciju informacija o ELABu i integraciju sa pojedinim ELAB servisima. Uzor može da bude MIT Android aplikacija https://play.google.com/store/apps/details?id=edu.mit.mitmobile2&hl=en

Aplikacija treba da obezbedi sledeće funkcionalnosti:

* Prikaz osnovnih informacija o ELABu: članovi, opis aktivnosti, kako do nas? kontakt i sl.
* Kalendar aktivnosti - Integracija sa Ajax calendar WP pluginom
* Integracija sa veb sajtom www.elab.rs - kada se ubaci nova vest, trebalo bi da se automatski prikaže i u aplikaciji
* Push notifikacije - nova vest, novi događaj
* Integracija sa društvenim mrežama
* Google material dizajn - Moderan dizajn po uputstvima na http://www.google.com/design/spec/material-design/introduction.html#introduction-principles

**Okvirni predlog sadržaja:**

1. Uvod
2. Analiza mobilnih aplikacija vodećih obrazovnih ustanova u svetu
3. Opis tehnologija primenjenihu razvoju rešenja
4. Razvoj Android/iOS ELAB aplikacija
5. Plan promocije
6. Zaključak
7. Literatura