**Prvi projektni zadatak / prva spirala**

*Rok za završetak: nedjelja, 29.3. u 23:59*

Potrebno je da napravite početni dizajn web stranice koristeći HTML i CSS (bez korištenja JavaScript).

Prvi projektni zadatak treba da poštuje sljedeću specifikaciju:

* Web stranica se treba sastojati od minimalno 4 podstranice (HTML datoteke). Podstranice trebaju biti povezane linkovima tako da se sa svake podstranice može doći na bilo koju drugu podstranicu (meni!)
* Sve podstranice trebaju imati unificiran dizajn tako da korisnik ima osjećaj da se nalazi na jednom web site-u.
* Za dizajn treba koristiti eksternu listu stilova (eksternu CSS datoteku!) Sve podstranice trebaju koristiti isti CSS. Unutar HTML datoteka ne smije se nalaziti STYLE tag niti STYLE atribut, odnosno sva definicija stila se treba nalaziti u eksternoj CSS datoteci. Također u HTML kodu se ne smiju nalaziti nikakvi tagovi niti atributi koji definišu izgled stranice (fontovi, boje...)
* Sve HTML i CSS datoteke trebaju biti validne (koristite [validator.w3.org](http://validator.w3.org/) i [jigsaw.w3.org/css-validator](http://jigsaw.w3.org/css-validator/))
* Stranice trebaju izgledati ~~identično~~ približno isto (bez vizuelnih artefakta i biti potpuno upotrebljive) na aktuelnim verzijama popularnih web browsera (Mozilla Firefox, Google Chrome, Internet Explorer) na Windows i Linux operativnim sistemima, te ispravno se prikazivati na svim često korištenim rezolucijama ekrana.
* Na svim podstranicama treba biti vidljiv logotip firme. Za logotip koristite Google Web fontove i neku sliku.
* Sav kod koji pošaljete (HTML, CSS) treba biti isključivo sastavljen od vašeg samostalnog rada! Nije dozvoljeno korištenje biblioteka gotovog CSS ili JavaScript koda (npr. Bootstrap). Nije dozvoljeno korištenje code snippets sa interneta. Nije dozvoljena razmjena koda sa drugim studentima. Nije dozvoljeno korištenje automatskih generatora koda (programa koji generišu CSS ili HTML kod na osnovu specifikacije). Svi projekti koji budu sadržavali takav kod biće bodovani sa 0 na ovoj spirali.

Unutar projekta treba se nalaziti barem po jedna stranica koja zadovoljava sljedeće uslove:

**NASLOVNICA**

Stranica koja je označena kao naslovnica treba obavezno sadržavati niz novosti. Svaka novost se sastoji od naslova, imena autora, datuma objave, teksta i linka sa tekstom "Detaljnije..." koji za sada ne treba da vodi nigdje. Organizacija stranice treba biti takva da se jednostavno mogu dodavati novosti i da se stranica neće deformisati bez obzira da li se na njoj nalazi jedna, dvije, pet ili stotinu novosti (vertikalni klizač se ne smatra deformisanjem, ali horizontalni se smatra!) Uz svaku vijest se može (ali ne mora) nalaziti i slika - dodajte barem jednu sliku. Pored ovoga dodajte još neki sadržaj na naslovnicu po vašem izboru.

**TABELARNI PRIKAZ PODATAKA**

Na jednoj od stranica (koja nije naslovnica niti kontakt forma) treba se koristiti HTML tabela za prikaz podataka koji su po svojoj prirodi tabelarni. Tabela treba imati zaglavlje, barem pet redova i barem tri kolone. HTML tabele se ne trebaju koristiti nizašta drugo osim za prikaz tabelarnih podataka! – DOKUMENTI TAB

**LINKOVI**

Na jednoj od stranica treba se nalaziti i spisak linkova na eksterne web stranice (srodne stranice, partneri i sl.) Linkovi trebaju biti predstavljeni u vidu neuređene liste (unordered list) gdje se umjesto tačke (bullet) prikazuje neka sličica po vašem izboru (vidjeti Tutorijal 2 Zadatak 2).

**KONTAKT FORMA**

Barem jedna stranica treba nuditi formular za kontakt vlasnika stranice (taj formular za sada neće biti funkcionalan pošto nismo još implementirali serversku stranu). Na formularu se trebaju nalaziti barem tri HTML5 kontrole. Zbog sporadične podrške za HTML5, formular treba izgledati potpuno ispravno na barem jednom od vodećih web browsera (trebate navesti kojem).

# Drugi projektni zadatak / druga spirala

## Validacija forme

Na neku formu (to može biti kontakt forma) trebate dodati JavaScript validaciju prema principima koji su dati na Tutorijalu 4. Formu ne treba biti moguće poslati ukoliko nije validna. Pored svakog polja koje ne zadovoljava uslove validnosti treba biti prikazan neki grafički element (npr. sličica sa uzvičnikom, X ili nešto slično) i prateći tekst koji objašnjava u čemu je greška.

* Forma treba sadržavati neka polja koja su obavezna i neka koja nisu.
* Za minimalno jedno polje se treba koristiti regex za validaciju.
* Za minimalno jedno polje treba koristiti validaciju koja nije zasnovana na regexu.
* Minimalno jedno polje na formi treba biti zavisno od drugog polja (npr. padajuća lista određuje opseg mogućih vrijednosti u drugom polju, cross-validacija i slično).

## Meni / stablo

Potrebno je na web stranici samostalno implementirati **u JavaScriptu** jednu od dvije kontrole po želji: padajući meni ili stablo.

**Padajući meni** treba funkcionisati po principu menija u desktop aplikacijama. Kada se klikne na neku od opcija u glavnom meniju treba se ispod nje otvoriti podmeni sa opcijama. Kada se mišem pređe preko neke od opcija u podmeniju ona treba promijeniti boju pozadine.

# Treći projektni zadatak / Treća spirala

*Rok za završetak: srijeda, 29.4. u 23:59*

**6 bodova**

*Napomena: Za rješavanje ovih zadataka, kao ni ranije, nije dozvoljeno korištenje bilo kakvih biblioteka, frameworka itd.*

***Važna napomena (dodato u ponedjeljak 27.4.): Pošto smo uočili da jedan broj studenata nije u potpunosti shvatilo tutorijal vezan za AJAX ili nisu istražili literaturu, poziv koji nije asinhron ne može se smatrati za AJAX, dakle ne priznaje se rješenje ovih zadataka u kojima treći parametar metode XMLHttpRequest.open() postavite na false.***

## Validacija forme putem AJAXa

Potrebno je dodati na formu validaciju putem AJAXa, na način da se validira pojedinačna vrijednost dva polja kao i da li dva polja odgovaraju. Npr. na formi imate polja "mjesto" i "općina", treba validirati da li postoji mjesto, da li postoji općina, te da li je dato mjesto u datoj općini. Validaciju treba obaviti po pravilima datim ranije. Za ovaj zadatak trebate iskoristiti jedan od sljedećih validacijskih servisa.

* [http://zamger.etf.unsa.ba/wt/mjesto\_opcina.php](http://zamger.etf.unsa.ba/wt/mjesto_opcina.php?opcina=Ilid%C5%BEa&mjesto=Ilid%C5%BEa)  
  Putem GET metode servis prima dva parametra: **opcina** i **mjesto**. Vraća se JSON objekat sa ključem "error" ako: općina ne postoji, mjesto ne postoji ili mjesto nije iz date općine, u suprotnom se vraća "ok".
* [http://zamger.etf.unsa.ba/wt/postanskiBroj.php](http://zamger.etf.unsa.ba/wt/postanskiBroj.php?mjesto=Sarajevo&postanskiBroj=71000)  
  Putem GET metode servis prima dva parametra: **mjesto** i **postanskiBroj**. Vraća se JSON objekat sa ključem "error" ako: poštanski broj ne postoji, mjesto ne postoji ili poštanski broj ne odgovara mjestu, u suprotnom se vraća "ok".
* <http://restcountries.eu/>  
  Razne informacije o državama (vidjeti dokumentaciju). Ovaj web servis možete iskoristiti da provjerite: da li telefonski pozivni broj odgovara državi, da li valuta odgovara državi, da li glavni grad odgovara državi, da li internet domena odgovara državi.
* [http://zamger.etf.unsa.ba/wt/srednja\_skola.php](http://zamger.etf.unsa.ba/wt/srednja_skola.php?opcina=Ilid%C5%BEa&skola=%C4%8Cetvrta%20gimnazija)  
  Putem GET metode servis prima dva parametra: **opcina** i **skola**. Vraća se JSON objekat sa ključem "error" ako: općina ne postoji, srednja škola ne postoji ili srednja škola nije iz date općine, u suprotnom se vraća "ok".
* [http://zamger.etf.unsa.ba/wt/predmet\_sifra.php](http://zamger.etf.unsa.ba/wt/predmet_sifra.php?predmet=Osnove%20ra%C4%8Dunarstva&sifra=ETF%20OR%20I-1170)  
  Putem GET metode servis prima dva parametra: **predmet** i **sifra**. Vraća se JSON objekat sa ključem "error" ako: predmet ne postoji, šifra predmeta ne postoji ili predmet nema tu šifru, u suprotnom se vraća "ok".

## Učitavanje podstranica putem AJAXa

Cilj je da vaša stranica postane [Single-page application](http://en.wikipedia.org/wiki/Single-page_application). Sve podstranice se trebaju učitavati putem AJAXa. Linkovi u meniju trebaju biti zamijenjeni pozivom funkcije koja vrši učitavanje odgovarajuće podstranice iz neke HTML datoteke na serveru, kao u Tutorijalu 5 Zadatak 1.

## Rad sa tabelom putem AJAXa

Dat je web servis:  
**http://zamger.etf.unsa.ba/wt/proizvodi.php**  
Ovaj web servis je vrlo sličan onome koji je opisan u posljednjem zadatku tutorijala 5, pa ću opisati samo razlike u odnosu na taj.

* Pri radu sa ovim web servisom *obavezno* uvijek morate uključiti parametar **brindexa** (cijeli broj) koji predstavlja vaš broj indeksa. Na taj način će svaki student imati svoje podatke.
* U bazi se nalaze određeni proizvodi. Svaki proizvod opisan je poljima: naziv, opis, slika, url (ovo su tekstualna polja proizvoljne dužine), kolicina (cijeli broj), cijena (realan broj), dostupnost (1 ili 0). Vi ne morate koristiti sva polja ako vam ne trebaju, jedino ustvari obavezno polje je naziv. Ako bi za vaš zadatak bilo korisno da se dodaju još neki tipovi polja - javite (obavještenje o tome će biti dodato na ovu stranicu).
* Na web stranici treba prikazati tabelu proizvoda sa slikama i opisom. Treba biti moguće dodavati, mijenjati i brisati proizvode (za sada ćemo tu mogućnost ostaviti svima, a kasnije se eventualno može prebaciti u administratorski dio). Forma za dodavanje novog proizvoda treba biti validirana (možete proglasiti još neka polja obaveznim po želji). Studentima je ostavljeno na volju da li će koristiti TABLE ili će pomoću CSSa i DIV/SPAN tagova organizirati prikaz.
* Sve spomenute operacije trebaju se izvoditi putem AJAXa. Nikada se ne smije desiti reload stranice.

# Četvrti projektni zadatak / Četvrta spirala

**Spirala nosi: 6 bodova**

**Napomena (12.5. u 10:30):** Nakon mnogobrojnih pitanja putem maila dodajemo pojašnjenje: u zadatku "Kontakt forma" nije obavezno pridržavati se SPA jer bi to znatno zakomplikovalo neke stvari imajući u vidu uslove zadatka. Ako baš želite da tako uradite - možete ali obratite pažnju da se nevalidna forma ne smije poslati bez obzira kako je korisnik manipulisao JavaScript kodom (npr. kroz konzolu i slično). Poenta validacije na serverskoj strani je da se klijentu ne može vjerovati, dok svrha validacije na klijentskoj strani je brži odziv, prilagođenost korisničkog interfejsa i slično.

U ovoj spirali trebate koristeći programski jezik PHP dodati vašoj web stranici funkcionalnosti navedene ispod. Nije dozvoljeno korištenje pomoćnih biblioteka, frameworka i sl. (Codeigniter, Laravel...) osim standardne PHP biblioteke. Sve izmjene trebate u datom roku postaviti na Github i hosting. Potrebno je propisno uraditi dekompoziciju koda tako da se izbjegava miješanje PHP, HTML, CSS i JavaScript koda kada god je to moguće (ponekad neće biti moguće).

Kod trebate dodati u web stranicu koju ste poslali u sklopu treće spirale. U principu nije dozvoljeno mijenjati kod poslan kao treća spirala osim radi popravke grešaka i realizacije zadataka datih ispod.

## Kontakt forma (3 boda)

Svrha ovog zadatka je da konačno stavimo u funkciju kontakt formu koja je dotjerivana tokom prošlih spirala. Vaša kontakt forma treba sadržavati **sva polja** koja je imala na drugoj spirali (ako ste u međuvremenu nešto izbacili - ponovo dodajte). Nije dozvoljeno brisanje ili promjena polja (osim radi popravke bugova), ali je dozvoljeno da dodate eventualno nova polja ako mislite da bi bila korisna.

Kada korisnik klikne na dugme za slanje forme, podatke treba poslati PHP skripti koja treba da obavlja sljedeće zadatke:

* Potrebno je validirati podatke na formi za slučaj da je korisnik isključio JavaScript ili da koristi browser koji ne podržava HTML5 (npr. IE8). PHP skripta treba raditi sljedeće:
  + Sva polja koja su označena kao obavezna ne smiju ostati prazna ili biti popunjena isključivo whitespace karakterima. Ako ranije niste imali obavezna polja na formi, dodajte barem jedno takvo polje.
  + Sva polja koja su tipa email trebaju imati format email adrese.
  + Sva polja koja su numeričkog tipa trebaju biti broj.
  + Potrebno je validirati i radio dugmad, padajuće liste itd. jer korisnik može pokušati prepraviti HTML kod prije slanja forme. Ova polja ne smiju imati nikakvu vrijednost osim dozvoljenih vrijednosti.
  + Sva polja koja se unakrsno validiraju (npr. "password" i "ponovi password") treba unakrsno provjeriti. Ako niste imali takva polja ranije, dodajte barem jedno polje sa unakrsnom validacijom.
  + Polja sa AJAX validacijom dodana u trećoj spirali ne treba provjeravati kroz PHP.
* U slučaju da forma ne prolazi validaciju, treba ponovo ispisati formu pri čemu sva polja forme trebaju biti popunjena istim vrijednostima koje je korisnik unio. Pored svakog neispravnog polja treba ispisati oznaku da je to polje neispravno (grafički element i/ili tekst) i to istu onakvu kakvu ste u prethodnim spiralama ispisivali kroz JavaScript.
* Ako je forma ispravna treba ispisati stranicu za potvrdu slanja koja izgleda ovako:
  + Najprije treba ispisati podnaslov "Provjerite da li ste ispravno popunili kontakt formu"
  + Ispod ovog podnaslova treba ispisati sve podatke koje je korisnik popunio u formi, na pregledan i jednostavan način (koristite CSS!)
  + Zatim treba ispisati poruku "Da li ste sigurni da želite poslati ove podatke?" i dugme na kojem stoji "Siguran sam". Kasnije ćemo objasniti šta ovo dugme radi.
  + Ispod toga treba ispisati podnaslov "Ako ste pogrešno popunili formu, možete ispod prepraviti unesene podatke".
  + Onda treba ispisati ponovo identičnu formu kao i ranije, koja treba biti popunjena podacima koje je korisnik unio tako da se mogu prepraviti greške. Treba ponuditi dugmad za slanje i reset kao i ranije. Ako korisnik klikne na dugme za slanje, ponovo dolazi na isti ovaj ekran za potvrdu sve dok ne klikne na dugme "Siguran sam".
* Kada se klikne na dugme "Siguran sam", treba putem maila poslati sadržaj forme na **vaš** mail. Poruka treba ispunjavati sljedeće:
  + Poruku na Cc: trebate poslati i svom tutoru (obratite pažnju da je ovo Cc: polje a ne da dodate još jednu adresu u To: polje).
  + Polje Reply-To: treba imati vrijednost email adresu koju je korisnik u email polje kontakt forme, tako da se može odmah odgovoriti pošiljaocu.
  + Poslani email treba imati naslov (subject) iz kojeg se može odmah prepoznati da je to poruka poslana sa kontakt forme.
  + Poslani email treba imati jasno i pregledno prikazan sadržaj forme koju je korisnik popunio, ali bez korištenja HTMLa (ako šaljete HTML mail onda morate imati i tekstualnu verziju za mail klijente koji ne podržavaju HTML).
  + Naša slova trebaju biti korektno prikazana u mailu.
  + Kada se mail pošalje, treba ispisati poruku "Zahvaljujemo se što ste nas kontaktirali".
* Sve stranice generisane ovom PHP skriptom trebaju koristiti isti izgled i dizajn kao ostatak sajta. Na svakoj stranici treba biti dostupan meni. Korisnik klikom na meni opciju "Kontakt forma" može ustvari poništiti do sada popunjene podatke.
* Nigdje u ovoj PHP skripti ne smije se javiti Cross-Site Scripting (XSS) propust!
* Izbjegavajte ponavljanje koda bilo koje vrste! Ako se u realizaciji ovog zadatka bude ponavljao kod (bilo da je to HTML, CSS, JavaScript ili PHP kod) tutori će oduzeti određeni (relativno mali) broj bodova.

*Ukoliko ste na vašem sajtu pravili neku drugu vrstu forme umjesto kontakt forme, potrebno je da ovaj zadatak prilagodite toj formi ili da dodate kontakt formu. Javite se tutoru za detalje!*

## Novosti (3 boda)

U ovom zadatku ćemo omogućiti vlasniku stranice da relativno lagano mijenja stranicu sa novostima. Stranicu sa novostima će ubuduće generisati PHP skripta koja treba da radi na sljedeći način:

U istom folderu gdje se nalazi PHP skripta treba se nalaziti pod-folder pod imenom "novosti". U tom folderu se trebaju nalaziti novosti u vidu niza TXT fajlova različitih imena koja nisu bitna (npr. "aaaa.txt", "proba.txt", "New document (1).txt" i slično).

Svaki fajl treba biti sljedećeg formata:

1. U prvom redu fajla se nalazi datum i vrijeme kreiranja novosti u formatu *dd.mm.yyyy. hh:mm:ss*.
2. U drugom redu se nalazi ime autora novosti.
3. U trećem redu se nalazi NASLOV NOVOSTI DAT VELIKIM SLOVIMA.
4. U četvrtom redu se nalazi URL slike. Ako nema slike, ovaj red je prazan.
5. U petom, šestom itd. redovima se nalazi tekst novosti. Tekst se završava krajem datoteke ili redom u kojem se nalaze isključivo dva znaka minus "--". Obratite pažnju da se unutar teksta novosti mogu nalaziti dva minusa, ali samo ako se ta dva minusa nalaze u zasebnom redu oni označavaju kraj teksta.
6. Ako se u datoteci nalazi red sa dva minusa, od sljedećeg reda iza njih pa do kraja datoteke nalazi se detaljniji tekst novosti. U suprotnom, za ovu novost ne postoji detaljniji tekst.

Datoteka je kodirana UTF-8 kodnim rasporedom i može sadržavati naša slova. Primjer sadržaja datoteke:

28.04.2015. 10:31:44

Vedran Ljubović

OVO JE NEKI PRIMJER NOVOSTI

https://zamger.etf.unsa.ba/images/16x16/zad\_ok.png

Sada ću napisati neki osnovni tekst.

Ovaj osnovni tekst se nalazi u više redova.

Lorem ipsum dolor sit amet i tako dalje mrsko mi je da kopiram.

--

Ovdje sada slijedi detaljniji tekst novosti.

Lorem ipsum dolor sit amet i tako dalje mrsko mi je da kopiram.

Lorem ipsum dolor sit amet i tako dalje mrsko mi je da kopiram.

Sve novosti u ovom folderu treba prikazati na web stranici u dijelu sa novostima. Novosti trebaju biti sortirane po datumu i vremenu objavljivanja, od novijih ka starijim. Naslov novosti treba prebaciti tako da je samo Prvo slovo veliko a ostala su mala. Ako je data slika, potrebno je prikazati sliku u za to predviđenom prostoru, u suprotnom ne.

Ako vijest ne sadrži detaljniji tekst, ne treba prikazati link Detaljnije. Ako sadrži, treba dati ovaj link tako da on vodi na PHP skriptu koja prikazuje tu vijest na zasebnoj stranici. Na ovoj stranici treba prikazati: naslov, ime autora, datum objavljivanja, sliku, tekst vijesti i detaljniji tekst vijesti (bez prikazivanja drugih vijesti itd.) Prikaz pojedinačne vijesti treba slijediti logiku SPA (single-page application) dakle klik na link Detaljnije ne smije izazvati reload stranice.

Nigdje u ovoj PHP skripti ne smije se javiti Cross-Site Scripting (XSS) propust! Također ne treba biti moguće otvoriti datoteke koje nisu novosti.

## Upload projekta na hosting platformu

U narednim danima dobićete pristupne podatke za web hosting vaših projekata. Potrebno je uraditi upload web stranice na hosting tako da je ona dostupna unosom linka koji vam je dat. Sve podstranice, linkovi itd. vezani za vašu stranicu trebaju biti funkcionalni kao i sve PHP skripte.

# Podaci za hosting

Skrećemo pažnju studentima da će se uspješan deployment web aplikacije ocjenjivati u sklopu četvrte spirale.

## OpenShift

1. Prijavite se na stranicu [www.openshift.com](https://www.openshift.com/)
2. Izaberite "Free" opciju
3. Prijavite se sa svojim zvaničnim fakultetskim mailom
4. Nakon konfirmacije izaberite "Create your first application now"
5. Izaberite PHP 5.4 cartridge
6. Unesite neki prepoznatljiv "Public URL", pod "Source Code" unesite podatke vašeg GitHub repozitorija
7. Kliknite na "Not now, continue".
8. Izaberite da dodate MySQL cartridge, pošto će vam trebati za spirale 5 i 6. Zapišite podatke za pristup mysql bazi koje će vam sistem izbaciti.
9. Radi lakšeg rada sa bazom preporučujemo da dodate i phpMyAdmin. U dnu ekrana imate opciju "Tools and Support" "Add phpMyAdmin 4.0" - izaberite i to!
10. Za daljnji rad sa OpenShiftom pročitajte uputstva (pod [Help / Getting Started](https://developers.openshift.com/en/getting-started-overview.html)). Slijedite uputstva za "Installing the Client Tools" i "Setting Up Your Machine" ali nemojte za "Creating Your Application" jer ste to već uradili kroz web interface. Ako ste dobro uradili korake trebalo bi na web interface da ugledate nešto kao:
11. Source Code
12. ssh://..../~/git/php.git/

Pass this URL to 'git clone' to copy the repository locally.

To je URL git repozitorija koji se koristi za upravljanje OpenShift stranicom. Ako ste ispravno popunili podatke ranije trebalo bi da ste dobili kopiju vašeg Github repozitorija. Kasnije kada nešto promijenite na githubu iskopirajte i ovdje.

1. Unesite naziv repozitorija koji ste kreirali u ovaj GDoc: [Hosting podaci](https://docs.google.com/a/etf.unsa.ba/document/d/1l_IzQbAIgLnY2IXAcH-XuOd3htsbxu0_bfzJ9pzlejo/edit?usp=sharing)

# Peti projektni zadatak / Peta spirala

*Rok za završetak: četvrtak, 28.5. u 23:59*

***Važna napomena:****Na openshift serveru i na githubu se mora nalaziti identičan kod! (Osim konfiguracije kao što su pristupni podaci za bazu) Sve uočene razlike između ova dva koda smatraće se pokušajem varanja i biti kažnjene sa 0 bodova.*

**Spirala nosi: 6 bodova**

Tema ove spirale je kombinacija PHP skripti sa MySQL bazom podataka i korištenje sesija. Pošto ovaj zadatak uključuje i bazu podataka **obavezno** trebate na svoj github repozitorij postaviti i bazu u vidu SQL dump-a. Ovaj dump dobijate na način koji je objašnjen na tutorijalu 8. Sve funkcionalnosti navedene u ovom zadatku moraju se nalaziti i na githubu i na [hosting platformi čije ste podatke dobili](http://c2.etf.unsa.ba/mod/resource/view.php?id=40803). PHP skripte i baza trebaju biti potpuno funkcionalne na hosting platformi, a na githubu se treba nalaziti sve što je potrebno da tutor može koristiti sajt na svom računaru. U slučaju da zadatak ispunite samo na jednom od ova dva mjesta dobijate 50% predviđenih bodova!

Sve funkcionalnosti trebaju biti osigurane od svih poznatih tipova sigurnosnih propusta (XSS, SQL injection...). Postojanje sigurnosnih propusta će biti kažnjeno velikim brojem negativnih bodova.

Dizajn baze treba slijediti pravila koja ste naučili na predmetu "Osnove baza podataka" - baza treba biti normalizovana i koristiti relacije tamo gdje to ima smisla.

Nije dozvoljeno korištenje pomoćnih biblioteka, frameworka i sl. (Codeigniter, Laravel...) osim standardne PHP biblioteke. Sve izmjene trebate u datom roku postaviti na Github i hosting. Potrebno je propisno uraditi dekompoziciju koda tako da se izbjegava miješanje PHP, HTML, CSS i JavaScript koda kada god je to moguće (ponekad neće biti moguće).

## Novosti u bazi (1 bod)

U prethodnoj spirali smo prikazivali novosti iz niza datoteka. Sada trebate novosti držati u MySQL bazi podataka. Polja koja opisuju jednu novost trebaju biti ista (naslov, tekst, autor, datum objave...).

## Komentari na novosti (2 boda)

Anonimni posjetitelji trebaju imati mogućnost ostavljanja poruka, bilo u vidu komentara na novosti ili negdje drugo na sajtu gdje ste to predvidjeli. Ispod svake novosti reba se prikazati broj komentara. Kada kliknete na taj link trebaju se prikazati svi komentari od starijih ka novijim. Svaki komentar treba sadržavati: datum i vrijeme ostavljanja komentara, autor komentara, email adresu autora i tekst komentara. Ime i email unosi komentator po želji, pri čemu polje email nije obavezno, a ako je unesen email klik na ime autora pokreće slanje maila toj osobi (mailto: link).

## Admin panel (3 boda)

Na nekoj lokaciji na sajtu treba se nalaziti administratorski prostor. Administrator se prijavljuje svojim korisničkim imenom i šifrom. Ako je administrator zaboravio šifru, treba postojati mogućnost da se šifra resetuje na neku random generisanu i zatim pošalje na administratorov email koji je negdje konfigurisan. Ako je uneseno ispravno korisničko ime i šifra, otvara se administratorski panel gdje su dostupne sljedeće opcije:

* Dodavanje, promjenu i brisanje novosti.
* Brisanje komentara na novosti radi uklanjanja nepristojnih komentara.
* Dodavanje, promjenu i brisanje korisnika koji imaju pristup administratorskom panelu (ovdje se unosi i email adresa koja je spomenuta iznad). Potrebno je spriječiti situaciju da ne postoji niti jedan korisnik koji ima pristup admin panelu.

# Šesti projektni zadatak / šesta spirala

*Rok za završetak: četvrtak, 11.6. u 23:59*

***Važna napomena:****Na openshift serveru i na githubu se mora nalaziti identičan kod! (Osim konfiguracije kao što su pristupni podaci za bazu) Sve uočene razlike između ova dva koda smatraće se pokušajem varanja i biti kažnjene sa 0 bodova.*

**Spirala nosi: 7 bodova**

U ovoj spirali ćemo prepraviti naš projekat tako da se koriste REST web servisi koje ćete razviti u programskom jeziku PHP kao što je obrađeno na tutorijalu 10. Pošto ovaj zadatak uključuje i bazu podataka obavezno trebate na svoj github repozitorij postaviti i bazu u vidu SQL dump-a. Ovaj dump dobijate na način koji je objašnjen na tutorijalu 8. Sve funkcionalnosti navedene u ovom zadatku moraju se nalaziti i na githubu i na hosting platformi čije ste podatke dobili. PHP skripte i baza trebaju biti potpuno funkcionalne na hosting platformi, a na githubu se treba nalaziti sve što je potrebno da tutor može koristiti sajt na svom računaru. U slučaju da zadatak ispunite samo na jednom od ova dva mjesta dobijate 50% predviđenih bodova!

Sve funkcionalnosti trebaju biti osigurane od svih poznatih tipova sigurnosnih propusta (XSS, SQL injection...). Postojanje sigurnosnih propusta će biti kažnjeno velikim brojem negativnih bodova.

Dizajn baze treba slijediti pravila koja ste naučili na predmetu "Osnove baza podataka" - baza treba biti normalizovana i koristiti relacije tamo gdje to ima smisla.

Nije dozvoljeno korištenje pomoćnih biblioteka, frameworka i sl. (Codeigniter, Laravel...) osim standardne PHP biblioteke. Za JavaScript također nije dozvoljeno koristiti pomoćne biblioteke, frameworke i sl. (jQuery, Angular, Bootstrap...) Sve izmjene trebate u datom roku postaviti na Github i hosting. Potrebno je propisno uraditi dekompoziciju koda tako da se izbjegava miješanje PHP, HTML, CSS i JavaScript koda (prelazak na web servise omogućuje da se ovaj zahtjev potpuno provede, pa ćete dobijati i negativne bodove ako ga ne ispunite). Lijepo organizirajte svoj kod po folderima npr. php folder, css folder, js folder.

## Postojeće funkcionalnosti prebaciti na web servise (3 boda)

Sve funkcionalnosti u programskom jeziku PHP koje ste na svom projektu trebali implementirati u spiralama 4 i 5 sada trebate prepraviti tako da se koriste web servisi. (Da, ovo znači da sve što niste uradili na prethodnim spiralama trebate uraditi sada u suprotnom ponovo gubite bodove. Izuzetak je što se novosti trebaju prikazivati samo iz baze a ne iz .txt fajlova.)

Sve PHP skripte trebaju se nalaziti u posebnom folderu. U PHP skriptama ne smije se generisati nikakav HTML niti JavaScript kod, isključivo JSON. JavaScript kod putem AJAXa poziva napravljeni web servis u PHPu, preuzima rezultate i prikazuje ih na ekranu. Krajnji korisnik ne smije vidjeti nikakve razlike u odnosu na petu spiralu osim onoga što je traženo u ovoj spirali kao nova funkcionalnost. Ako studenti žele dodati neke nove funkcionalnosti u projekat trebaju javiti tutoru i prodiskutovati kako to utiče na postavku zadatka.

Kada završite ove zadatke stari kod vezan za spirale 4 i 5 ne smije ostati na repozitoriju niti na hostingu.

Obratite pažnju na sljedeće detalje:

* Validaciju kontakt forme treba obavljati web servis, kao i slanje maila (isti ili neki drugi), ali ni na koji način ne smije biti moguće poslati nevalidan mail uključujući i slučaj korisnika koji ukucava neke maliciozne JavaScript komande ili prepravlja sadržaj stranice koristeći Developer tools alate i slično.
* Za ovaj zadatak prikaz detaljnijeg teksta vijesti i komentara možete obaviti na bilo koji način (ne mora striktno biti druga PHP skripta kao u 5. spirali), ali pogledajte i sljedeći zadatak koji dodaje neka ograničenja po tom pitanju.
* Kod prijavljivanja na admin panel ponašanje treba biti isto kao na 5. spirali odnosno kako smo obradili na tutorijalu. Korisnik treba ostati prijavljen na sajt sve dok se ne odjavi (ili dok ne istekne vrijeme za sesiju), pa čak i ako recimo zatvori web preglednik i zatim opet otvori i ukuca direktno adresu u adresnu traku.
* SPA implementaciju treba doraditi tako da je Back dugme preglednika funkcionalno (vraća se na prethodnu podstranicu vašeg sajta a ne tamo gdje ste bili prije nego što ste otvorili sajt).
* Sve ostalo što je prirodno i logično da radi i što se očekivalo da radi na ranijim spiralama biće bodovano.

## Dinamičko učitavanje (2 boda)

Ovaj zadatak se nadovezuje na prethodni. Kada se unese nova vijest ili novi komentar, oni se trebaju prikazati svim korisnicima koji trenutno imaju otvorenu stranicu. Korisnici ne trebaju raditi reload da bi ugledali novu vijest ili komentar! Bukvalno ako otvorite dva različita preglednika koji prikazuju vašu stranicu, kada ukucate komentar u jednom pregledniku on se treba pojaviti na drugom pregledniku.

To ćete postići tako što će JavaScript periodično provjeravati ima li novih vijesti ili novih komentara i u skladu s tim ažurirati sadržaj stranice. Pri tome osigurajte da preglednik nije vidno usporen, da nije opterećen mrežni link korisnika, a da se nove izmjene na stranici prikažu što brže. Ako admin edituje vijest (promijeni tekst vijesti) ili obriše vijest, ili ako obriše komentar, sve to se treba dinamički prikazati.

Da bi prikaz vijesti i komentara bio zaista dinamičan, treba prikaz komentara obaviti na sljedeći način:

* Razdvojiti linkove "Detaljnije" i "X komentara"
* Kada korisnik klikne na "Detaljnije" kontaktira se web servis i preuzima se detaljniji tekst novosti koji se zatim prikazuje korisniku tako što se ubacuje ispod osnovnog teksta vijesti. Treba biti dostupna i opcija "Sakrij detaljnije".
* Broj X u linku "X komentara" treba se dinamički ažurirati (čim neko doda novi komentar).
* Kada se klikne na link "X komentara" opet treba prikazati te komentare ispod tekuće vijesti, bez reloada, promjene ostatka stranice itd. (U 5. spirali se tražilo da se prikaže samo ta vijest bez ostalih, a ovdje više nije tako.) Treba biti dostupna opcija "Sakrij detaljnije". Čim neko doda novi komentar on se treba ubaciti sve dok su vidljivi komentari.
* Kada je aktivno prikazivanje komentara, ispod postojećih komentara treba biti ponuđena i forma za ostavljanje novog komentara.

## Korisnici na čitavom sajtu (1 bod)

U spirali 5 korisnički nalozi koje ste pravili odnosili su se samo na admin sekciju. Sada trebate proširiti tu funkcionalnost na ostatak sajta.

Forma za prijavu na sajt treba biti vidljiva na svim stranicama sajta (ako korisnik nije prijavljen), a ako jeste treba biti vidljivo ime korisnika i dostupna opcija za odjavu.

Prilikom ostavljanja komentara, polja za ime autora i adresu ne trebaju više biti slobodna polja za unos. Kolona u bazi koja označava autora komentara treba biti strani ključ (foreign key) na tabelu korisnika. Pored toga treba biti moguće ostavljati anonimne komentare. U slučaju da korisnik nije trenutno prijavljen, komentar koji on ostavlja označava se kao anoniman. Link mailto treba biti popunjen email adresom koju ste podesili u bazi.

Potrebno je da postoje dva tipa korisnika: obični i admin korisnici. Samo admin korisnici imaju pravo pristupa admin panelu koji ste napravili u spirali 5, a obični mogu samo komentarisati vijesti. I dalje važi pravilo da na sajtu mora postojati barem jedan admin korisnik (ali ne mora postojati nijedan obični korisnik).

## Resetovanje passworda (1 bod)

Opcija za resetovanje šifre putem maila koju ste napravili na spirali 5 sada također treba biti dostupna svima, s tim što ćemo je malo doraditi: Onako kako smo definisali u spirali 5 neki zlonamjerni posjetilac bi vam mogao neprekidno resetovati šifru. Zbog toga se šifra neće automatski resetovati kada se izabere opcija "Zaboravljena šifra". Umjesto toga, korisniku treba poslati konfirmacijski mail sa nekim linkom, pa tek kada korisnik klikne na taj link dolazi do resetovanja šifre. Link za resetovanje šifre treba važiti 24 sata od slanja zahtjeva za resetovanje šifre.

(Ako ste sve to napravili na spirali 5 ne morate to ponovo raditi na spirali 6).

**Upute za ocjenjivanje sveukupnog dojma aplikacije**

Pod kategorijom "sveukupni dojam" ocjenjivaće se koliko je web stranica koju ste napravili blizu nečemu što bi vaš klijent (realni ili imaginarni) zaista želio koristiti na svojoj zvaničnoj adresi. Ocjenjivaće se stanje na githubu/openshiftu na dan 15.6.2016. Dakle imaćete po završetku 6. spirale još par dana da iz projekta uklonite neke stvari koje ste dodali radi zadataka (spirala) a smatrate da nisu bitne klijentu imajući u vidu temu stranice. Možete dodati i neke druge stvari koje smatrate da su klijentu interesantne. Ako vam za dodavanje tih drugih stvari treba još vremena, možete se obratiti tutoru za produženje roka (imajte na umu termine za završetak ispita i upis ocjene).

Ocjenjivaće se sve ono što je bitno za jednu web stranicu, uključujući:

* dizajn i izgled stranice
* funkcionalnost stranice (da li je dostupno sve ono što je potrebno korisniku)
* odsustvo vidljivih grešaka bilo kojeg tipa
* upotrebljivost (eng. usability, pogodnost za korištenje)
* sigurnost
* itd.

Sve one greške koje su napravljene u ranijim spiralama ulaze u ocjenu ali i neke greške koje nisu prijavljene. Argument "niste nam rekli da je to greška" se ne uvažava.