

Razvoj web aplikacija

Projektna specifikacija

1. Opće informacije

- Obrana rješenja projektnog zadatka održava se tijekom ispitnih rokova.
- Student prijavljuje obranu kao i ostale ispite.
- Važna pravila za uspješnu predaju rješenja projektnog zadatka:
 - Predajte projektni zadatak na vrijeme
 - Poslana zip arhiva mora slijediti specificirano imenovanje i strukturu datoteke
 - Vaše rješenje neka ima barem minimalne bodove po cilju učenja
 - Vaše rješenje mora pratiti temu koju je odobrio profesor

Predaja zadatka na vrijeme

- **5 dana prije obrane** smatra se rokom za dostavu rješenja projektnog zadatka. Rješenje projektnog zadatka potrebno je dostaviti najkasnije u roku. Ovdje slijedi primjer objašnjenog rasporeda. U primjeru je određen datum obrane 9.2.

...	3.2.	4.2.	5.2.	6.2.	7.2.	8.2.	9.2.
		Rok 23:59					Obrana
<i>Rješavanje projektnog zadatka da se postigne barem minimum po ishodu</i>			<i>Poboljšanje rješenja za postizanje više od minimuma</i>				

- Studentima koji predaju rješenje projektnog zadatka nakon roka neće biti prihvaćen rad i neće moći pristupiti obrani.

Naziv i struktura datoteke poslane zip arhive

- Projektni zadatak u obliku **zip** arhive pošaljite na e-mail adresu svog profesora/asistenta. RAR, 7z i drugi formati arhiva neće biti prihvaćeni.
- Oblik naziva arhive mora biti **{prezime}-{ime}-{naziv Vašeg projekta}.zip**
Primjer: Smith-John-MyProjectTask.zip
- Struktura arhive mora slijediti objašnjenje iz **sekcije 3**. Neispravna struktura arhive neće biti prihvaćena. **Molim da pročitate sekciju 3!**

Minimalni broj bodova po ishodu

- Ispravno pripremljeno rješenje projektnog zadatka nosi barem minimalni broj bodova po cilju učenja, odnosno 10 bodova za svaki cilj učenja.
- Pazite da ispunite barem minimalni dio za svaki ishod, inače ćete biti automatski diskvalificirani na obrani.

Poboljšanje vašeg rada i povratne informacije

- Nakon isteka roka, a prije dana obrane, studenti mogu doraditi projektni zadatak i ponovno ga predati.
- Studenti mogu očekivati povratnu informaciju o svom predanom radu prije isteka roka. U nekim slučajevima studenti mogu dobiti povratnu informaciju nakon roka i prije obrane, ali to nije zajamčeno.

2. Specifikacija projektnog zadatka

Napravite jedinstveno ASP.NET Core web rješenje (engl. *solution*) koje se sastoji od dva modula (projekta).

Prilikom izrade rješenja slijedite temu koju ste odabrali i koju je profesor odobrio.

Postoje dva modula koja trebate implementirati:

- RESTful service modul (Web API)
 - Pokriva I1 i I2
 - Koristi se za dohvaćanje podataka putem JavaScripta putem npr. statične HTML stranice. Također namijenjen za automatizaciju (na primjer prikazivanje ili ažuriranje podataka o unosu video sadržaja putem API-ja).
- MVC modul (Web aplikacija)
 - Pokriva I3, I4 i I5
 - Korisnik pristupa putem web preglednika

2.1. Ishod 1 (RESTful service modul, Web API)

Minimum (10 bodova): Stvorite RESTful krajnju točku (CRUD) za vaš primarni entitet, s krajnjim točkama za pretraživanje i straničenje (engl. *paging*). Pišite u zapisnik (engl. *log*) tijekom izvođenja ovih operacija i učinite te zapisnike dostupnima putem dodatne krajnje točke. **Željeno (10 bodova):** Osigurajte svoje krajnje točke pomoću autentifikacije JWT tokenom i implementirajte uobičajene autentifikacijske funkcije.

Općenite odrednice, minimum (10 bodova)

Za CRUD krajnje točke **primarnog entiteta** upotrijebite naziv primarnog entiteta (npr. *api/video*). Koristite JSON sadržaj u tijelu zahtjeva gdje je to prikladno. Važno je obraditi pogreške i u HTTP odgovorima vratiti kodove pogrešaka 400, 404 i 500 gdje je to potrebno. Podržite dodatnu krajnju točku za funkcionalnost pretraživanja prema npr. atributima *Name*, *Description* ili oboje. Ova krajnja točka također mora podržavati funkcionalnost straničenja koristeći parametre npr. *Page* i *Count*. Za funkciju pretraživanja upotrijebite odgovarajući naziv krajnje točke (npr. *api/video/search*).

Za vaše **zapisnike** (engl. *logs*) implementirajte odgovarajuću krajnju točku (npr. *api/logs/get/N* - vraća zadnjih N zapisnika, gdje se N prosljeđuje parametrom usmjeravanja, engl. *routing parameter*). Implementirajte krajnju točku *api/logs/count* koja vraća ukupan broj spremljenih zapisa. Za pojedinačni zapis koristite atribute npr. *Id*, *Timestamp*, *Level* i *Message*. Zabilježite u zapisnik svaku CRUD akciju svog primarnog entiteta, kao smislenu poruku koja identificira što se dogodilo s instancom entiteta. Primjeri: "*Video sadržaj s id=7 je stvoren.*", "*Došlo je do problema prilikom ažuriranja video sadržaja s id=7.*" ili "*Ne mogu pronaći žanr id=11.*"

Provjerite jeste li uključili Swagger ili slično sučelje kako biste tijekom obrane mogli na jednostavan način demonstrirati kako vaš API radi.

Općenite odrednice, željeno (10 bodova):

Implementirajte JWT token autentifikaciju za krajnje točke **zapisnika** koje ste implementirali za minimum ishoda 1. Implementirajte dodavanje novog korisnika (npr.

endpoint `api/auth/register`), dohvaćanje JWT tokena za korisnika (npr. endpoint `api/auth/login`) i promjenu korisničke lozinke (npr. endpoint `api/auth/changepassword`).

Provjerite podržava li vaše Swagger sučelje autentifikaciju kako biste mogli jednostavno pokazati kako vaša autentifikacija funkcionira.
Možete također dodati token autentifikaciju drugim krajnjim točkama ako to želite.

2.2. Ishod 2 (RESTful service modul, Web API)

Minimum (10 bodova): Implementirajte pristup bazi podataka za svoje krajnje točke.

Željeno (10 bodova): Implementirajte statične HTML stranice koje koriste JWT autentifikaciju, localStorage i postojeće implementirane krajnje točke za siguran prikaz željenog broja zapisa dnevnika.

Općenite odrednice, minimum (10 bodova)

Koristite bazu podataka za spremanje stanja koristeći RESTful krajnju točku (CRUD) za vaš primarni entitet koji ste implementirali za minimum ishoda 1. Dodatno, implementirajte CRUD krajnje točke za entitete 1-na-N i M-na-N, za koje također podržite bazu podataka.
Obratite pažnju na brisanje povezanih entiteta i elegantno postupajte s eventualnim pogreškama.

Općenite odrednice, željeno (10 bodova)

Implementirajte statične HTML stranice koje koriste JWT autentifikaciju i postojeće krajnje točke za siguran prikaz zapisnika (engl. *logs*). Naime, te stranice bi trebale biti stranica za prijavu i stranica s popisom zapisnika (engl. *log list*). Na stranici s popisom zapisnika korisnik bi trebao moći promijeniti prikazani broj zapisa, npr. zadnjih 10, 25 ili 50 zapisa, ovisno o tome što je odabrano odgovarajućim padajućim izbornikom. Koristite localStorage za pohranjivanje autentifikacijskog tokena. Podržite odjavu klikom na gumb "Odjava".

Za pojedinosti o HTML stranicama možete vidjeti *wireframe*:

- Image 1 –
- Image 2 –

2.3. Ishod 3 (MVC modul, Web aplikacija)

Minimum (10 bodova): Za administratora, izradite sigurno web mjesto (engl. *website*)

koje implementira CRUD funkcionalnost za svaki od entiteta. Implementirajte smislenu i dosljednu navigaciju. **Poželjno (10 bodova):** Za korisnika izradite vizualno privlačno web mjesto s *landing* stranicom i fokusom na primarni entitet. Implementirajte smislenu i dosljednu navigaciju. Korisnik mora imati mogućnost samostalne registracije i prijave. Korisnik treba stranicu na kojoj može vidjeti popis željenih stavki, mora imati način da otvori željenu stavku i moći obaviti željenu radnju kao što je gledanje filma, rezerviranje ulaznice, prijava za natjecanje, dodavanje proizvoda u košaricu itd. Za ulogu administratora također implementirajte prikaz popisa korisnika sa njihovim željenim radnjama, poput gledanih filmova, rezerviranih ulaznica, prijavama za natjecanje ili sadržaja košarice za kupnju.

Općenite odrednice, minimum (10 bodova)

Implementirajte sljedeće za administratora:

1. **Login stranica:** Uspješna prijava vodi na List stranicu primarnog entiteta.
2. **CRUD stranice za primarni entitet CRUD:** List, Add, Edit i Delete stranice primarnog entiteta.
Na **List** stranici treba postojati tekstni okvir za pretraživanje i padajući izbornik od stavki 1-na-N entiteta za filtriranje stavki. Filtriranje se vrši kada kliknete gumb Search. Klikom na gumb Previous ili Next dolazi se do prethodnih ili sljedećih 10 stavki.
3. **CRUD stranice za ostale entitete:** List, Add, Edit i Delete stranice za 1-na-N entitet i M-na-N entitet.

Stranice moraju sadržavati smislenu i dosljednu navigaciju. Preciznije: sve spomenute stranice osim stranice za prijavu moraju sadržavati navigaciju koja vodi do stranica s popisom (engl. *list page*) za primarne, 1-na-N i M-na-N entitete; također, svaka stranica treba imati gumb za odjavu.

Prikažite stranice na vizualno privlačan način.

Općenite odrednice, željeno (10 bodova)

Implementirajte sljedeće za korisnika:

1. **Landing stranica:** Nije potrebno unijeti pristupne podatke za pristup *landing* stranici. *Landing* stranica vizualno predstavlja vašu temu. *CTA* vodi do stranice za prijavu.
2. **Stranica za samostalnu registraciju:** Korisnik može unijeti podatke za registraciju (korisničko ime, e-mail, lozinka, ponovljena lozinka...) i samostalno se registrirati.
3. **Login stranica:** Ovisno o ulozi (engl. *role*), uspješna prijava vodi do npr. List stranice primarnog entiteta kod administratora ili kod korisnika.
4. **Stranica sa stavkama:** Korisniku prikazuje popis stavki primarnih entiteta. Trebao bi postojati tekstualni okvir za pretraživanje (na primjer, naziv filma) i padajući izbornik za 1-na-N entitet za filtriranje (na primjer, žanr filma). Filtriranje se vrši kada kliknete gumb Search. Sa stranice sa stavkama klikom na svaku stavku dolazi se na stranicu s detaljima (na primjer, informacije o filmu). Klikom na gumb Previous ili Next prelazi se na prethodnih ili sljedećih 10 stavki.
5. **Stranica s detaljima:** Prikaz atributa primarnog entiteta. Omogućite korisniku da se vrati na stranicu sa stavkama.
6. **Željena akcija:** Korisnik mora moći izvršiti željenu akciju na stranici s detaljima o određene stavke (gledanje filma, rezervacija ulaznice, prijava na natječaj, dodavanje proizvoda u košaricu za kupnju itd.)

Implementirajte sljedeće za administratora:

7. Podrška za prikaz popisa korisnika i njihovih željenih radnji, poput filmova koje je korisnik gledao, rezerviranih ulaznica, prijava za natječaje ili sadržaja njezine/njegove košarice za kupnju.

Prikažite stranice na vizualno privlačan način.

Podržite *upload* slike za primarni entitet ako to tema zahtijeva.

Pogledajte sekciju *Dodatak* za predložene sheme i *wireframe* za izgled HTML stranice. Sheme i *wireframe* samo su primjeri.

- Image 3 -
- Image 4 -

- Image 5 - Login
- Image 6 - *Landing*
- Image 7 - Stranica za popis primarnog entiteta (varijanta 1)
- Image 8 - Stranica za popis primarnog entiteta (varijanta 2)
- Image 9 - Stranica za detalje primarnog enti
- Image 10 -

2.4. Ishod 4 (MVC modul, Web aplikacija)

Minimum (10 bodova): Izvršite validaciju modela i labeliranje pomoću anotacija modela.

Željeno (10 bodova): Implementirajte smisleno višeslojno (engl. *multi-tier*) rješenje i u tom rješenju koristite AutoMapper za pojednostavljenje mapiranja modela.

Općenite odrednice, minimum (10 bodova)

Modeli moraju biti validirani: obavezna polja, ispravni URL-ovi, ispravne e-mail adrese itd. Implementirajte validaciju koja sprječava prazan unos (npr. Name, Description...) Duplicirani nazivi instanci entiteta nisu dopušteni (npr. dva žanra s nazivom "Thriller" ili dva filma s nazivom "Die Hard"). Vidljive labele moraju se implementirati pomoću anotacija modela. Identifikatori instance entiteta (ili *id-evi*) ne smiju biti vidljivi nigdje na korisničkom sučelju.

Općenite odrednice, željeno (10 bodova)

Iskoristite višeslojni koncept kako biste pojednostavili strukturu rješenja. Ideja je na kraju imati Web API i MVC slojeve (projekte) koji ovise o istom zajedničkom poslovnom sloju (projektu) i istom sloju baze podataka. *Imajte na umu da to znači da ćete na kraju imati više od dva projekta u svom rješenju.*

Modeli: trebali biste imati drugačiji skup modela za svaki *tier* (kojeg predstavlja projekt). Na primjer, model baze podataka ne smije se koristiti u prikazu. Ne biste trebali imati svojstva navigacije (engl. *navigation properties*) u modelima prikaza. Koristite AutoMapper za mapiranje modela između *tier-ova*.

2.5. Ishod 5 (MVC modul, Web aplikacija)

Minimum (10 bodova): Za administratora, implementirajte profilnu stranicu za ažuriranje njenih/njegovih osobnih podataka pomoću AJAX zahtjeva. **Poželjno (10 bodova):** Za korisnike, implementirajte profilnu stranicu za ažuriranje svojih osobnih podataka pomoću AJAX zahtjeva. Omogućite korisniku izvođenje kompleksnog navigacijskog straničenja na stranici s popisom primarnih entiteta pomoću AJAX zahtjeva.

Općenite odrednice, minimum (10 bodova)

Za administratora implementirajte **profilnu stranicu**. Administrator bi trebao moći promijeniti e-mail, ime, prezime, broj telefona i druge osobne podatke.

Morate koristiti AJAX zahtjeve u implementaciji.

Općenite odrednice, željeno (10 bodova)

Za korisnika implementirajte **profilnu stranicu**. Korisnici bi trebali moći promijeniti e-mail adresu, ime, prezime, broj telefona i druge osobne podatke.

Morate koristiti AJAX zahtjeve za implementaciju.

Implementirajte AJAX straničenje na **stranici sa stavkama za primarni entitet**.

Najbolji rezultat bio bi prikazati nekoliko stranica prije i poslije trenutne stranice (brojevi poput **5, 6, 7, 8**) i gumb Previous i Next.

Pogledajte sekciju *Dodatak* za *wireframe* izgleda HTML stranice:

1. Image 11 -

3. Struktura projektnog zadatka

Da bi vaš projekt bio prihvaćen, mora slijediti **ovdje opisanu strukturu**.

Struktura se sastoji od strukture arhive rješenja projektnog zadatka, strukture entiteta i strukture baze podataka koju stvara SQL skripta.

3.1. Struktura arhive rješenja projektnog zadatka

Arhiva rješenja projektnog zadatka mora biti ZIP arhiva sa sljedećom strukturom mape/datoteka:

- ProjectTask
 - Database
 - *Database.sql* (jedina SQL datoteka u mapi)
 - *SolutionName* (mapa nazvana prema temi)
 - *SolutionName.sln* (jedina datoteka rješenja u mapi, nazovite je prema temi, npr. *VideoContentManager.sln*)
 - WebAPI (vaša mapa s jednim Web API projektom *WebAPI.csproj* i njegovim datotekama)
 - WebApp (vaša mapa s jednim projektom Web aplikacije *WebApp.csproj* i njegovim datotekama)

Primjer strukture arhive možete pronaći na Infoeduki (*ProjectTask-example.zip*).

3.2. Struktura entiteta

Rješenje teme koje ste dogovorili s profesorom treba imati definirane entitete. Ti entiteti su sljedeći:

- **Primarni entitet** (na primjer, za temu Sustav upravljanja videosadržajem, glavni entitet bi bio **Video**)
- **Dodatni entiteti**
 - **1-prema-N entitet**: na primjer, **Genre**
 - **M-prema-N entitet**: na primjer, **Tag**; Relacija M-prema-N podrazumijeva postojanje tablica entiteta i tablica premošćivanja (engl. *bridge*) u bazi podataka, kao što su **Tag** i **VideoTag**
 - **Entitet korisnika aplikacije**: na primjer, **User**

- **Image entitet** (željeno, prema potrebi teme): na primjer, **Image**; taj entitet se koristi za predstavljanje slike primarnog entiteta
- **Korisnikov M-prema-N entitet** (željeno): *bridge* tablica koja bilježi željene radnje korisnika, poput akcije za dodavanje filma, rezervacije ulaznice, prijave na natječaj, dodavanja proizvoda u košaricu itd.

Svaki entitet mora imati atribut *Name*. Primarni entitet mora imati najmanje 3 dodatna druga atributa osim *Name* i *Id* (npr. *Duration*, *Description*, *VideoUrl*).

Sve tablice moraju imati isti naziv kao i njihovi entiteti. Na primjer, ne smijete tablicu imenovati *Video* i koristiti je kao entitet *Product* (npr. u projektu web trgovine).

Sva imena tablica moraju biti u jednini.

3.3. Struktura SQL skripte za izradu baze podataka

1. Datoteka skripte za izradu baze podataka obavezan je zahtjev. Zato dio rješenja koji se odnosi na bazu podataka mora slijediti načelo *database-first*. To je u suprotnosti s *code-first* načelom, koji uključuje migracije i koji se ne smije koristiti.
2. Sav kôd za kreiranje tablice baze podataka mora biti u datoteci *Database.sql* u arhivi.
3. **Nemojte koristiti modificiranje baze podataka (ALTER DATABASE), stvaranje (CREATE DATABASE) ili prebacivanje između baza podataka (USE) naredbe u datoteci Database.sql.**
4. Koristite naredbe za modificiranje tablice (ALTER TABLE) i kreiranje tablice (CREATE TABLE) u datoteci *Database.sql*, tamo gdje su vam potrebni.
5. Koristite naredbe za umetanje, modificiranje, brisanje i dohvaćanje slogova tablice u datoteci *Database.sql*, tamo gdje su vam potrebni
Imajte na umu da ćete tijekom obrane morati demonstrirati kako radi Vaša aplikacija. To znači da ćete morati imati neke vrijednosti za entitete 1-to-N i M-to-N u aplikaciji koja je instalirana na računalu u učionici u kojoj se odvija obrana.

Primjer skripte za izradu baze podataka možete pronaći u strukturi arhive na Infoeduki. Imajte na umu da skripta za izradu baze podataka sadrži primjere naziva tablica i stupaca, a ne stvarna imena koja su Vam potrebna u Vašem projektu.

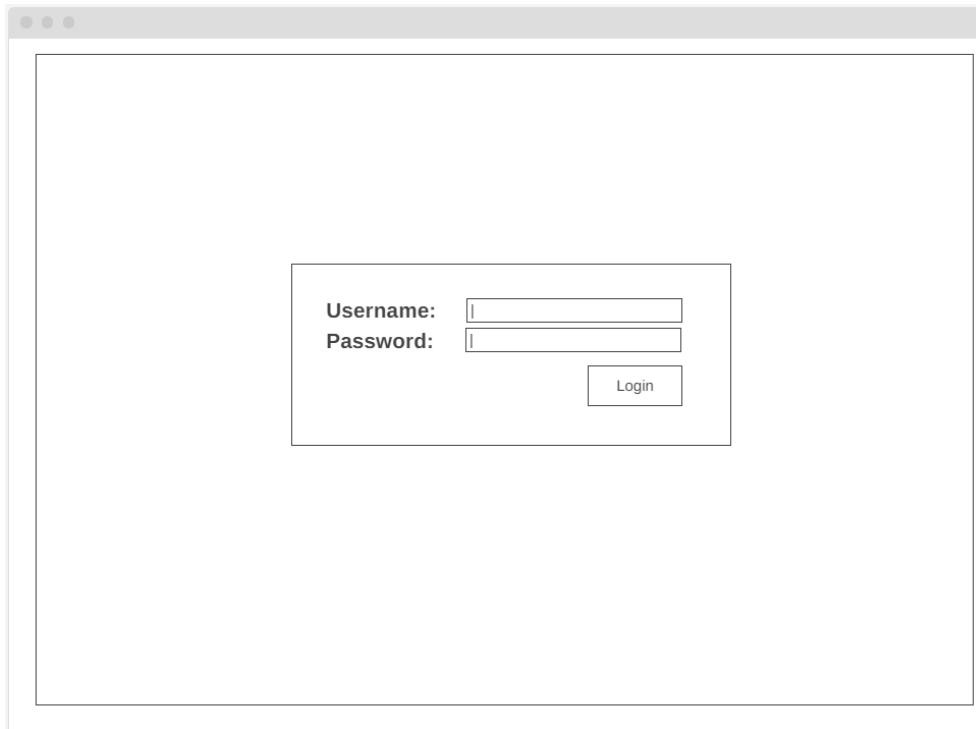
3.4. Važne napomene

1. Za format arhive koristite ZIP, RAR, 7z i ostali formati datoteke arhiva nisu podržani i neće biti prihvaćeni.
2. Arhiva ne smije sadržavati mape **bin** i **obj** zbog ograničenja antivirusnih e-mail filtera. Isto vrijedi i za druge mape koje sadrže artefakte (na primjer **packages**). Obavezno ih izbrišite prije stvaranja ZIP arhive. Imajte na umu da morate zatvoriti Visual Studio kako biste izbjegli automatsko ponovno stvaranje ovih mapa.

Ako je vaša ZIP arhiva prevelika (> 10 MB i slično) ili dobijete pogrešku prilikom slanja arhive putem e-maila, trebali biste provjeriti sadrži li arhiva bin, obj, packages ili druge slične mape s artefaktima.

3. **Hardkodiranje connection stringa je zabranjeno.** Connection string mora biti učitani iz konfiguracije (appsettings.json). Napominjem da jednostavno dodavanje connection stringa u datoteku appsettings.json ne rješava problem, morate također referencirati konfiguraciju iz koda.
4. Koristite **specificiranu .NET verziju** za razvoj. To bi trebala biti ista verzija koju ste koristili u svojim radionicama i zadacima. Ako koristite drugu verziju .NET-a, Vaša aplikacija se ne može instalirati i provjeriti automatskom provjerom i ne možete dobiti povratnu informaciju prije obrane.

4. Dodatak: okvirni izgled i raspored HTML stranica



A screenshot of a web browser window showing a login form. The form is centered and contains two input fields: 'Username:' and 'Password:'. Below the password field is a 'Login' button. The browser window has three dots in the top-left corner.

Image 1 – Statička Login stranica



A screenshot of a web browser window showing a 'Log list' page. The page has a title 'Log list' and a 'Logout' button in the top right. Below the title is a list of ten entries, each represented by a horizontal bar. At the bottom right, there is a dropdown menu showing '25' and a 'Show Logs' button. The browser window has three dots in the top-left corner.

Image 2 – Statička stranica s popisom logova

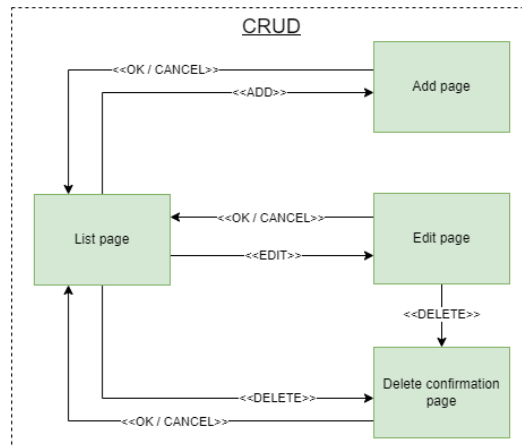


Image 3 - Shema za CRUD web stranice

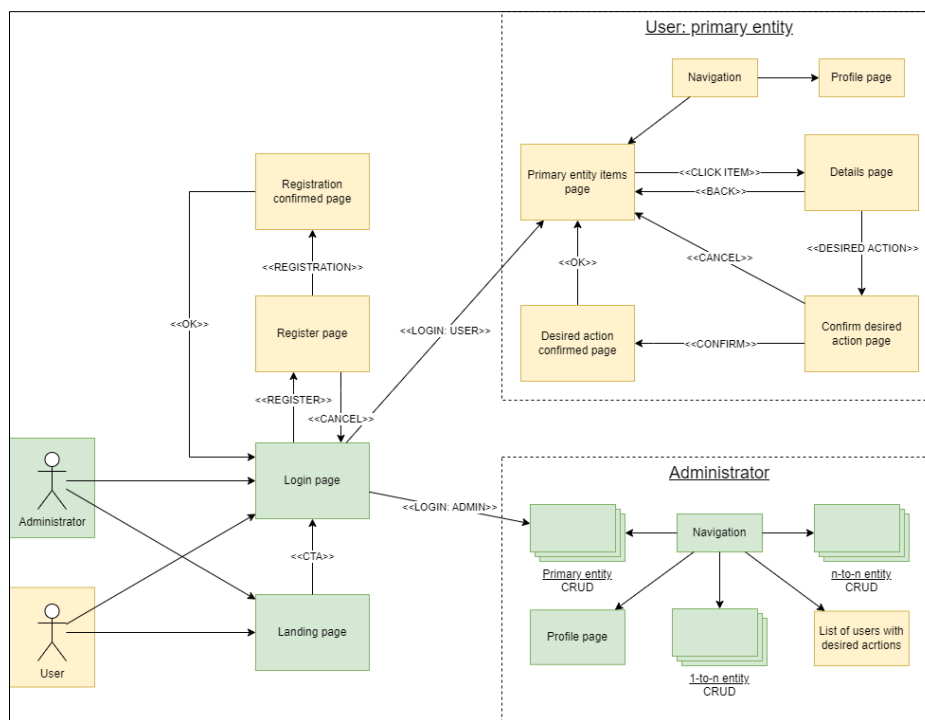


Image 4 - Navigacijska shema web aplikacije

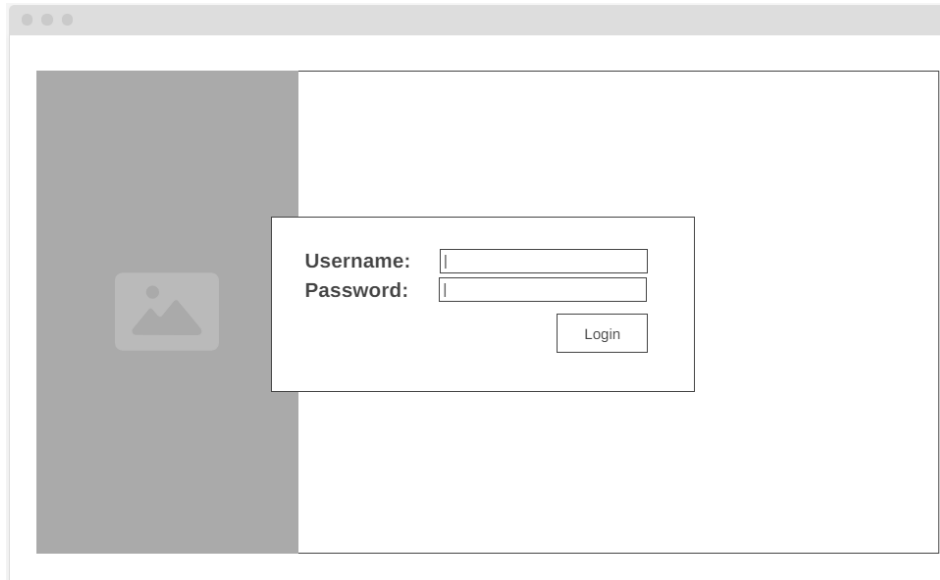


Image 5 - Login stranica

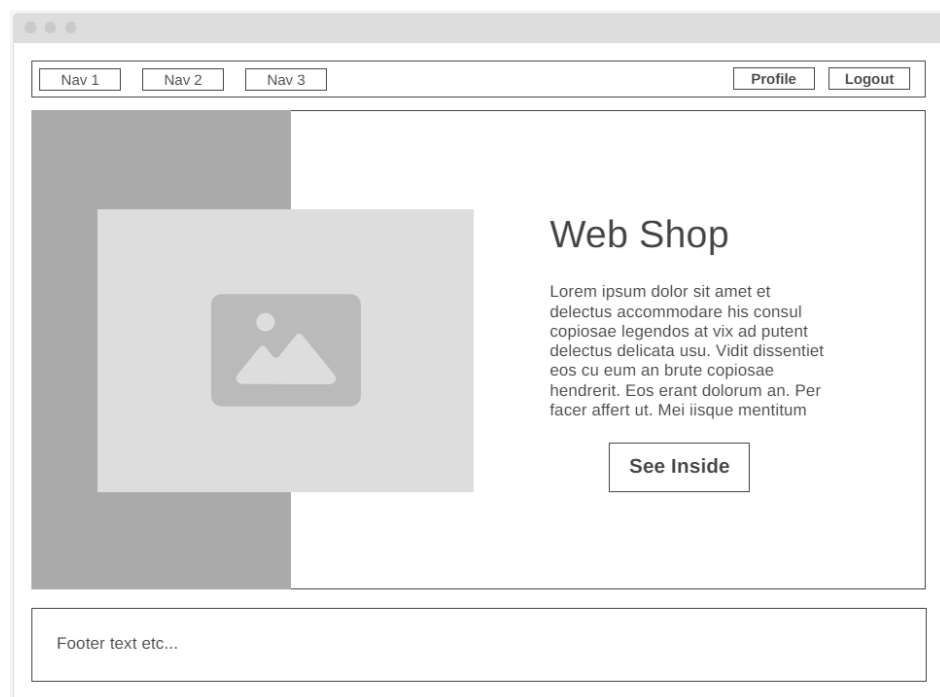


Image 6 - Landing stranica

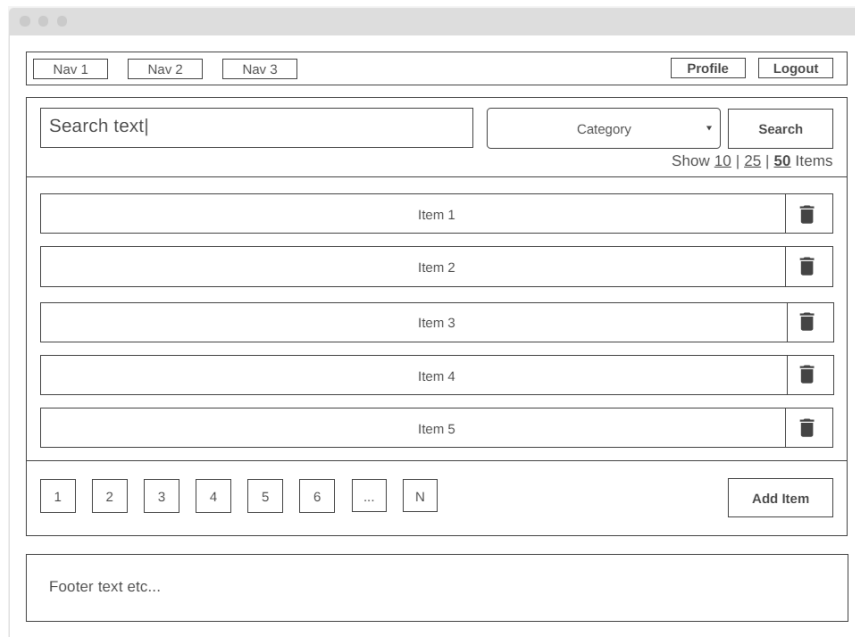


Image 7 – Stranica za popis primarnog entiteta (varijanta 1)

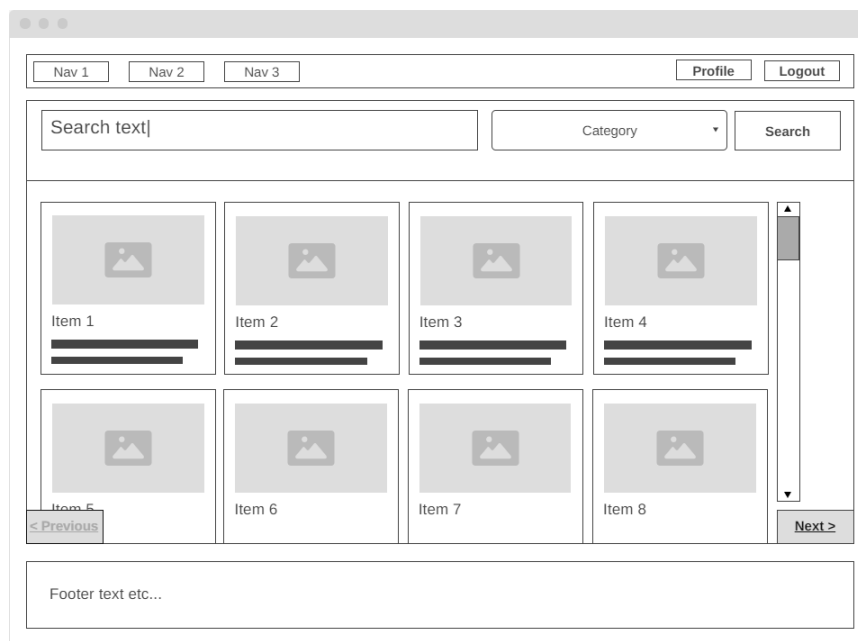


Image 8 - Stranica za popis primarnog entiteta (varijanta 2)

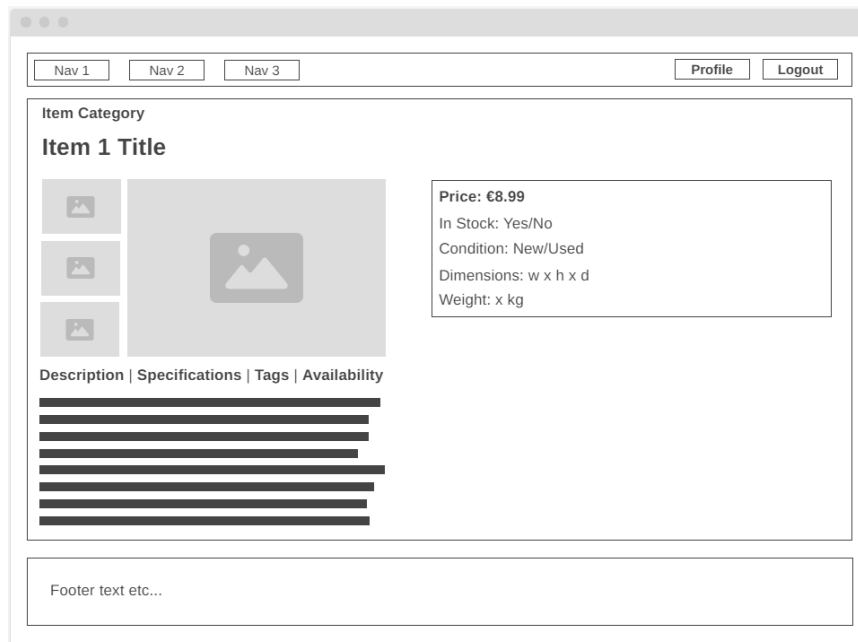


Image 9 – Stranica za detalje primarnog entiteta

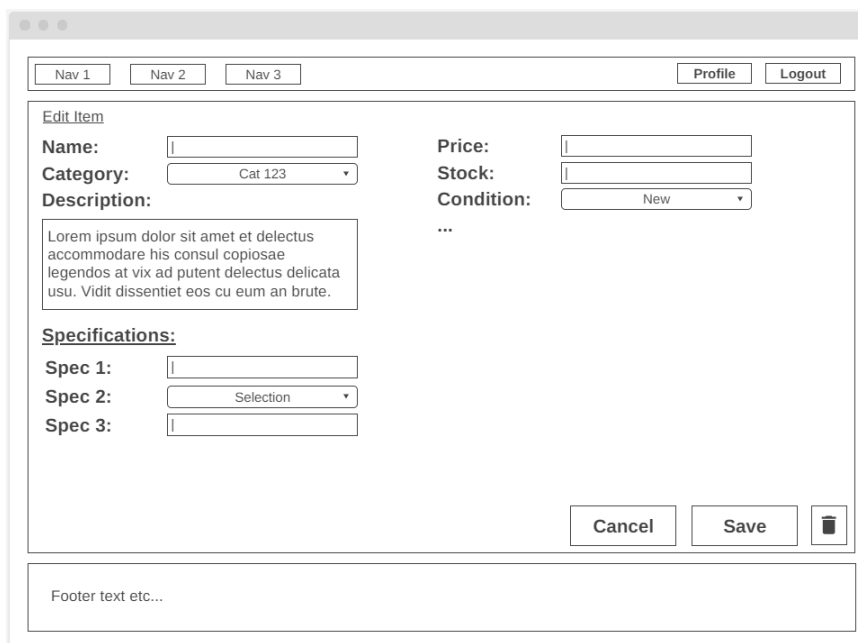
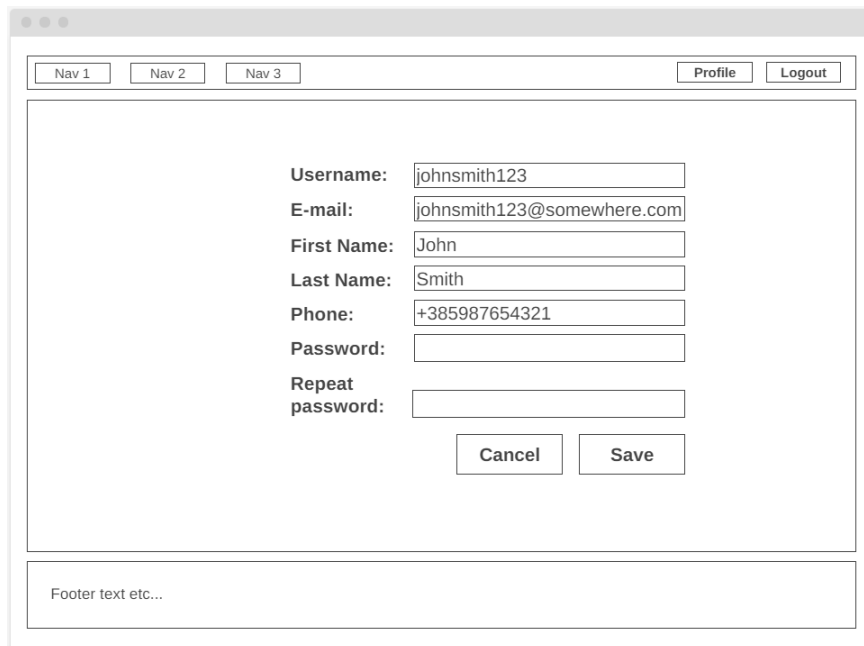


Image 10 - Add/Edit stranica za primarni entitet



The image shows a web browser window displaying a user profile page. At the top, there are three navigation tabs labeled 'Nav 1', 'Nav 2', and 'Nav 3'. To the right of these tabs are two buttons: 'Profile' and 'Logout'. The main content area contains a form with the following fields and values:

- Username:** johnsmith123
- E-mail:** johnsmith123@somewhere.com
- First Name:** John
- Last Name:** Smith
- Phone:** +385987654321
- Password:** (empty field)
- Repeat password:** (empty field)

Below the form are two buttons: 'Cancel' and 'Save'. At the bottom of the page, there is a footer area containing the text 'Footer text etc...'.

Image 11 - Profilna stranica