C# aplikacija za generiranje testova

Ivić, Vedran

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:682471

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij elektrotehnike

C# APLIKACIJA ZA GENERIRANJE TESTOVA

Završni rad

Vedran Ivić

Osijek, 2017.

SADRŽAJ

1. UVOD 1
1.1. Aplikacije za generiranje testova 1
1.2. Zadatak završnog rada 2
1.3. Opis postupka realizacije završnog rada 3
2. RESURSI, ALATI I TEHNOLOGIJE 4
2.1. Microsoft Visual Studio 4
2.2. Microsoft SQL Server 5
2.3. C#, ASP.NET, HTML i Entity Framework 6
3. BAZA PODATAKA 10
3.1. ER Dijagram 10
3.2. Relacijski model 11
3.3. Realizacija SQL upitima 12
4. APLIKACIJA 14
4.1. Povezivanje s bazom 14
4.2. Ključne datoteke 15
4.2.1. Datoteka *Baza_pitanja.aspx* 18
4.2.2. Datoteka *Generiraj.aspx* 21
4.2.3. Datoteka *Baza_testova.aspx* 25
4.3. Uređivanje stranica i ostale datoteke 28
5. ANALIZA I VREDNOVANJE (*USER EXPERIENCE*) 32
5.1. Izgled aplikacije 32
5.2. Testiranje 36
5.2.1. Generiranje nasumičnog testa 36
5.2.2. Generiranje testa osobnim odabirom i baza pitanja 38
5.2.3. Baza testova 44
6. ZAKLJUČAK 46
LITERATURA 47
SAŽETAK 48
C# APPLICATION FOR GENERATING TESTS 49
ŽIVOTOPIS 50
PRILOZI 51

1. UVOD

Ovaj rad podrazumijeva konceptualno i stvarno rješenje zadanog zadatka završnog rada. Naime, potrebno je realizirati C# aplikaciju za generiranje testova – web aplikacija koja bi po parametrima zadanim od strane korisnika generirala test (kontrolni testovi, ispiti, upitnici, kvizovi, ankete i sl.) na temelju pitanja iz postojeće baze podataka. Istu je potrebno dizajnirati i implementirati, uz osmišljavanje potrebnih tablica za bazu podataka koje će aplikacija koristiti. Aplikacija bi trebala omogućavati rad s pitanjima različite vrste – zaokruživanje točnog odgovora, pitanja s višestrukim izborom, pitanja kratkog i dugog odgovora, točno/netočno i slično. Korisnik bi trebao imati uvid i pristup bazi podataka u kojoj može mijenjati, uklanjati ili dodavati pitanja. Također, aplikacija treba omogućiti korisniku da samostalno odabire pitanja za test ili da to automatski radi nasumičnim odabirom u skladu sa zadanim kriterijima. Kao konačan rezultat korisnikovog rada s aplikacijom, potrebno je dobiti generiran željeni test.

Radi se, dakle, o realizaciji tzv. „stvaratelj kvizova“ (eng. *quiz builder*) tipa aplikacije odnosno aplikacije za generiranje testova koja za cilj ima automatizaciju te skraćivanje vremena izrade raznih vrsta testova. Stoga, takve aplikacije nalaze primjenu u širokom spektru djelatnosti (obrazovne ustanove, anketiranje, online kvizovi, testiranja znanja, vještina i sl.).

1.1. Aplikacije za generiranje testova

Različiti testovi i upitnici predstavljaju glavni oblik provjere određenog znanja i prikupljanja podataka o istom. Budući da je takve provjere najčešće potrebno provoditi više puta, ali s različitim uzorkom, od velike je koristi automatsko generiranje i izrada testova. U tu svrhu, javlja se potreba za razvojem aplikacija za generiranje testova koje bi uvelike pomogle pri testiranjima istog gradiva koje se periodično ponavlja i to na način da znatno olakšavaju i skraćuju vrijeme izrade testa ili upitnika koristeći već gotovu bazu pitanja te omogućavanjem automatizacije cijelog procesa. Primjena aplikacije može biti realizirana kod osoba koje provode ankete ili upitnike u svrhu prikupljanja podataka za istraživanja ili pak kod nastavnika i profesora za sastavljanje različitih provjera znanja.

Postoje određene online verzije i aplikacije za izradu testova (npr. *Testmoz, EasyTestMaker* i sl.), međutim sva su postojeća rješenja bazirana na jednokratnom kreiranju testa te pristupaju svim trenutnim pitanjima u bazi.

Ono što razlikuje ovu aplikaciju od prethodno spomenutih postojećih aplikacija je upravo selektivni proces pitanja te način rada s bazom pitanja. Naime, aplikacija omogućava odabir točno određenih pitanja prema zadanim kriterijima (npr. pitanja koja se nisu pojavila u zadnja dva testa ili samo pitanja čija je učestalost pojavljivanja u određenom rasponu). Dakle, aplikacija pruža znatno složeniji i funkcionalniji rad s bazom pitanja, što za rezultat ima kvalitetnije testove i provjere znanja koje su jednostavno generirane u kratkom vremenskom roku. Uz to, za krajnje efikasno sastavljanje testa, omogućeno je slučajno generiranje i odabir skupa pitanja, što znatno skraćuje proces izrade testa. Osim toga, omogućen je uvid o učestalosti pojedinih pitanja koja se pojavljuju kao i osobni odabir svakog pitanja pojedinačno ako korisnik to želi. U svakom trenutku pitanja je moguće pregledati, uređivati, dodavati, brisati, a za jednom generiran test, aplikacija omogućava pregled te ispis ili izvoz istog u datoteku, prilagođenog izgleda i dizajna. Parametri koji bi je trebali razlikovati od sličnih rješenja su efikasnost grafičkog sučelja, brzina izrade testa, veličina i raznolikost baze te pouzdanost odabranog testa (broj ponovljenih pitanja u odnosu na broj ponovljenih testova).

Ono što jedna takva aplikacija za generiranje testova mora sadržavati je, prvenstveno, baza podataka (pitanja, testova, parametara i sl.), zatim sučelje web aplikacije koje je *user-friendly* i jednostavno za korištenje te određen način povezivanja baze podataka sa samom aplikacijom i sučeljem. Svojstva koje takva aplikacija mora zadovoljiti, pored zadovoljavajućeg, intuitivnog i jasnog grafičkog prikaza, su zadovoljavajuća brzina izrade testa, odgovarajuća veličina baze te pouzdanost rezultata odnosno generiranog testa.

Nastavno na spomenuta svojstva odnosno kriterije i uvjete koje sam zadatak završnog rada zahtijeva, za realizaciju baze podataka, kao prvog aspekta same aplikacije, potrebna je programska podrška za rad s bazama (Microsoft SQL Server i Microsoft SQL Server Management Studio), a zatim za izradu web aplikacije i sučelja – programsko okruženje za razvoj web aplikacija i povezivanje s bazom (Microsoft Visual Studio za C# aplikaciju s ASP.NET i Entity Framework tehnologijama).

1.2. Zadatak završnog rada

Napraviti aplikaciju za automatsku izradu testova prema korisnički zadanim parametrima koristeći programski jezik C#. Na temelju korisnički zadanih parametara aplikacija treba generirati i po potrebi ispisati test sastavljen od pitanja koja se nalaze u bazi podataka. Primijeniti različite oblike pitanja (odabir točnih odgovora, esejska pitanja, nadopuna rečenice i sl. Također dizajnirati

i implementirati odgovarajuću bazu podataka u kojoj će biti pohranjena pitanja. Potrebno je omogućiti izmjenu, brisanje i dodavanje pitanja u bazu podataka, pregled generiranog testa te funkcionalnost za osobni ili slučajni odabir pitanja. Aplikacija treba biti primjenjiva za sastavljanje jednostavnih provjera znanja, anketa, testova osobnosti, kvizova i slično.

1.3. Opis postupka realizacije završnog rada

Na osnovu zadanog zadatka završnog rada te odabranih alata i tehnologija u skladu s istim, pristupilo se rješavanju zadatka i realizaciji aplikacije. Ponajprije je dizajnirana i implementirana baza podataka (pitanja, testova i parametara) s odgovarajućim tablicama, zatim je na osnovu iste, uz pomoć korištenih tehnologija, napravljen model podataka, potrebne datoteke i konfiguracije za povezivanje s bazom te u konačnici realizirana sama web aplikacija uz omogućavanje svih funkcionalnosti aplikacije te dizajn sučelja.

U poglavljima koja slijede opisani su ponajprije korišteni alati, tehnologije i potrebni resursi za izradu završnog rada uz bitnu teorijsku podlogu za pisanje istog, zatim je opisan koncept i realizacija baze podataka, potom sama aplikacija, njeni dijelovi i faze nastajanja odnosno proces izrade te u konačnici sučelje i izgled gotove aplikacije, realizirani rezultat te analiza istoga. Pri kraju samog rada, dan je osvrt na cjelokupan rad i postignute rezultate.

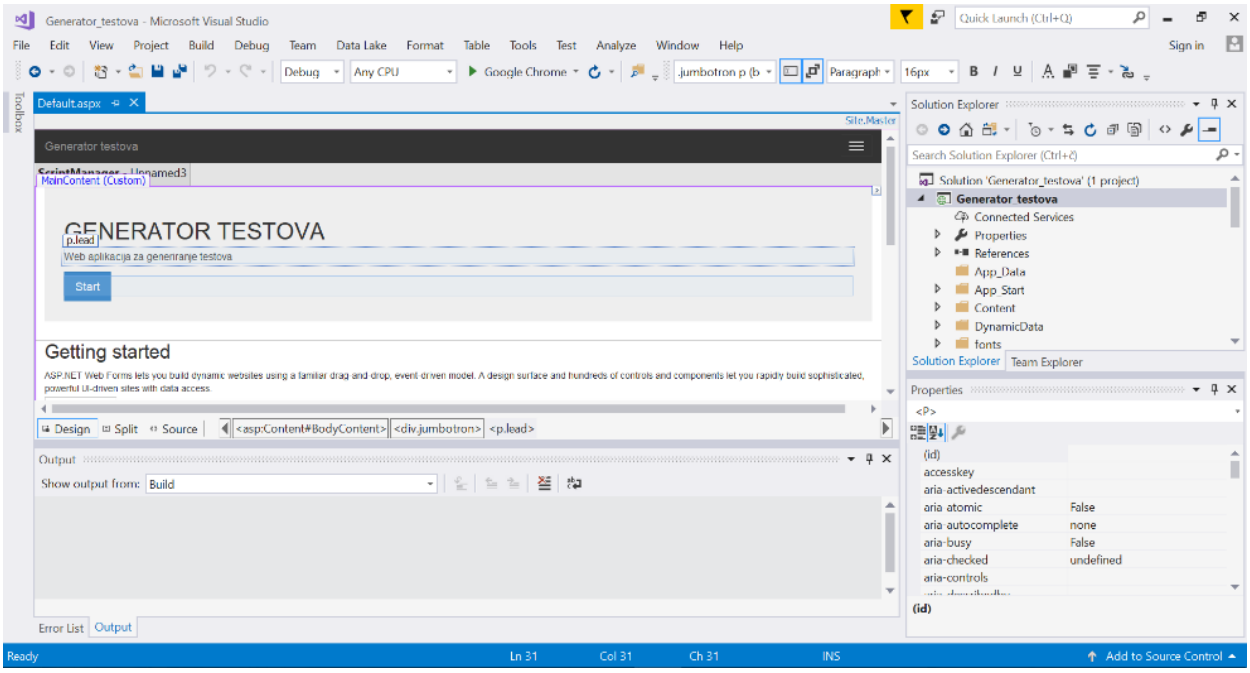
2. RESURSI, ALATI I TEHNOLOGIJE

Kao što je prethodno spomenuto, za izradu završnog rada na osnovu svojstava aplikacije i zahtjeva zadatka, korišteni su programski paketi: Microsoft Visual Studio 2017 (za razvoj aplikacije), Microsoft SQL Server 2016 (za realizaciju baze podataka), Google Chrome (za testiranje aplikacije) te programski jezici i tehnologije C#, ASP.NET, CSS, JavaScript, HTML te Entity Framework.

2.1. Microsoft Visual Studio

Microsoftov programski paket Visual Studio integrirano je razvojno okruženje namijenjeno razvoju aplikacija, programa za operativni sustav Windows, web stranica, web i mobilnih aplikacija i ostalih usluga, kao što je definirano u [1]. Za izradu ovog završnog rada, korištena je Visual Studio 2017 inačica programa koja je za studente Fakulteta elektrotehnike, računarstva i informacijskih tehnologija dostupna za besplatno preuzimanje u sklopu Microsoftovog DreamSpark programa.

Glavni dijelovi Visual Studio razvojnog okruženja su *code editor* (pisanje i uređivanje koda), *debugger* (pronalaženje grešaka u kodu/aplikaciji) te *designer* (dizajn formi, stranica, sučelja aplikacije, shema podataka i sl.). Sučelje samog programa organizirano je u nekoliko dijelova (prozora): *Solution Explorer* (prikaz datoteka projekta i projektnog rješenja), *Server Explorer* (prikaz za povezivanje sa serverom i bazama podataka), *Properties Window* (prozor s prikazom svojstava objekata), *Toolbox* (objekti i alati za dizajn aplikacije), *Toolbar* (alatne trake programa), te *Document Window* (prozor za prikaz sadržaja pojedinih datoteka). Navedeni dijelovi programa vidljivi su na slici 2.1., gdje je prikazano sučelje samog programa.



Sl. 2.1. *Izgled sučelja programa Visual Studio 2017*

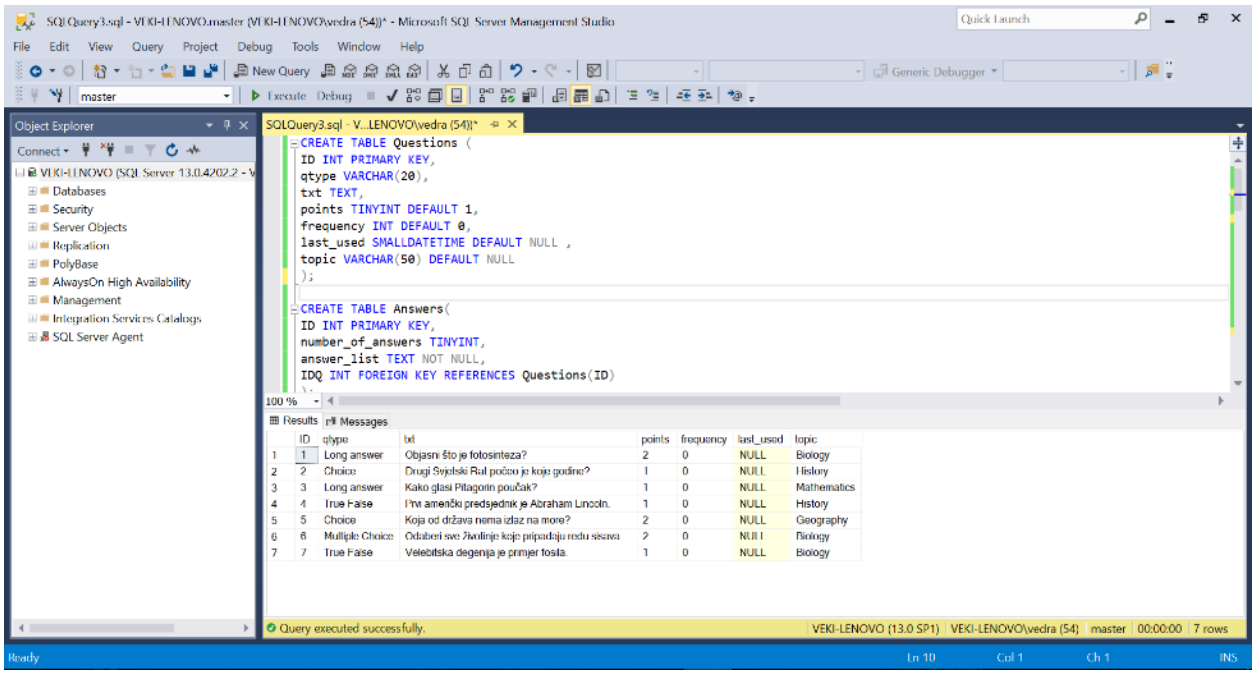
Visual Studio podržava rad u više ugrađenih programskih jezika i tehnologija kao što su C, C++, C#, Visual Basic, F# te HTML, JavaScript, CSS i slično. Prednost korištenja ovakvog programskog okruženja je prvenstveno velika adaptiranost i orijentiranost ka C# programskom jeziku te između ostalog i sva svojstva koja olakšavaju pisanje aplikacije u istom (npr. *IntelliSense* tehnologija koja omogućava predviđanje i sugestije za dijelove koda).

2.2. Microsoft SQL Server

Kao alat za rad s bazom podataka, odnosno dizajnom i realizacijom iste, odabran je Microsoft SQL Server programski paket. Prema [2], Microsoft SQL Server je relacijski sustav upravljanja bazom podataka s namjenom pohrane i pristupa podacima prema zahtjevima drugih aplikacija. Sam se sustav, dakle, temelji na relacijskom modelu baze podataka odnosno tablicama koji je u širokoj upotrebi, a radi sa strukturiranim jezikom upita (eng. *Structured Query Language*, SQL) – programskim jezikom za cjelokupno upravljanje podacima u relacijskim bazama podataka).

Za izradu ovog završnog rada, korištena je SQL Server 2016 inačica programa, a za sam rad s bazom podataka program Microsoft SQL Server Management Studio – upravljačko sučelje za rad sa SQL Serverom odnosno za upravljanje, administriranje i podešavanje svih komponenti koje SQL Server sadrži. Microsoft SQL Server Management Studio podrazumijeva grafičke alate

za rad s objektima na serveru te uređivanje skripti, a jedan od glavnih dijelova samog programa je *Object Explorer* koji omogućava pretragu, odabir i rad s objektima na serveru. Po uspostavljenoj vezi sa serverom, u spomenutom se *Object Exploreru* uočavaju potrebni objekti za rad, prvenstveno baze podataka (kreirane tablice, pogledi, procedure i sl.), ali i objekti za sigurnost na serveru, prava pristupa bazi podataka, vjerodajnice i slično. Samo sučelje sastoji se od alatnih traka programa, spomenutog *Object Explorera*, prozora za pisanje i uređivanje upita te prozora za prikaz rezultata i izlaza istih. Izgled sučelja prikazan je slikom 2.2.



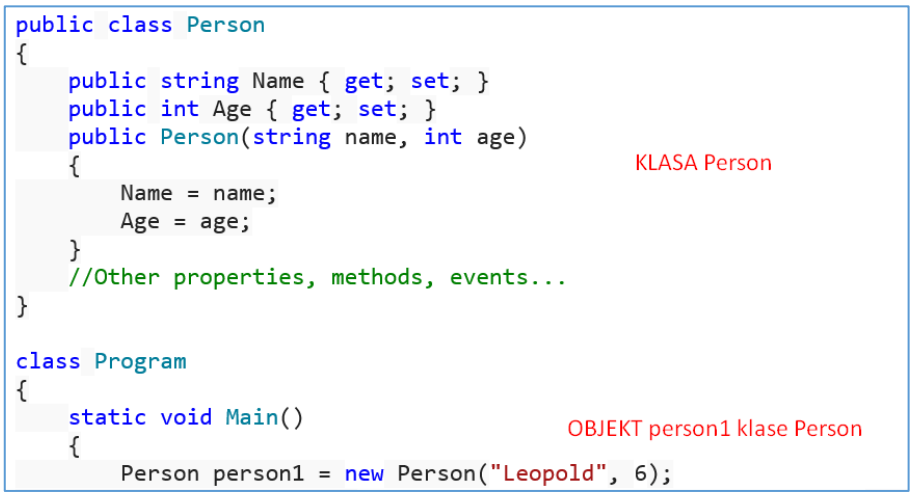
Sl. 2.2. *Izgled sučelja programa Microsoft SQL Server Management Studio*

2.3. C#, ASP.NET, HTML i Entity Framework

Odabrani programski jezik za razvoj web aplikacije je upravo **C#** (*C sharp*). To je objektno-orijentirani, općenamjenski programski jezik razvijen od strane Microsofta te osmišljen za razvoj i programiranje različitih aplikacija koje koriste i rade na .NET platformi (sam .NET objašnjen je u nastavku). Oslanja se na prethodne C programske jezike (velik dio sintakse, jednostavnost i izražajnost, objektno-orijentirana struktura, visoke performanse i sl.), ali se uvode i mnoge inovacije i značajke kojima se pospješuje brz i siguran razvoj aplikacija, kako je i objašnjeno u [3].

Neke od tipičnih značajki C#-a, prvenstveno kao objektno-orijentiranog jezika, prema [4], su hijerarhija klasa i objekata, enkapsulacija, nasljeđivanje te polimorfizam. **Klasa** ili razred

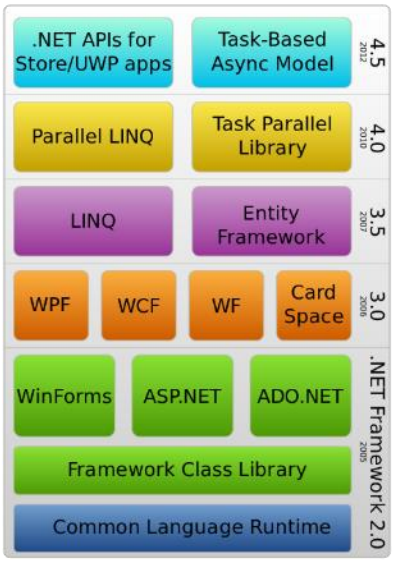
(*class*) predstavlja nacrt odnosno opis svojstava za skup objekata, a sastoji se od varijabli i metoda koje određuju i opisuju te objekte. **Objekt** predstavlja skup svojstava objedinjenih u smislenu cjelinu, odnosno konkretnu i specifičnu realizaciju klase u memoriji računala. Slika 2.3. prikazuje primjer klase i objekta.



Sl. 2.3. *Primjer klase (Person) i objekta (person1)*

Enkapsulacija objekta podrazumijeva opis javnog sučelja i implementaciju. Uz to, bitna svojstva za objektno-orjentirano programiranje su naslijeđivanje i polimorfizam. **Nasljeđivanjem** se osigurava specijalizacija, tj. uporaba svojstava već stvorenih klasa kod novih sličnih klasa, ali uz pojavu nekih specifičnih svojstava samo za te klase. Time se postiže hijerarhija klasa, što smanjuje količinu pisanog koda. **Polimorfizam** podrazumijeva stvaranje operacija koje su neovisne o vrsti, gdje se svaki objekt izvedene klase može smatrati objektom bilo koje javne osnovne klase.

Spomenuti ***.NET Framework*** predstavlja Microsoftovu razvojnu platformu odnosno softversko okruženje (okvir, *framework*) kod koga se softver s općim, generičkim funkcionalnostima može specificirati odnosno selektivno izmijeniti kodom od strane korisnika, što za rezultat daje aplikacije specifične i željene namjene (web, Windows, Windows Phone, Windows Server, Microsoft Azure i sl.). Kroz godine, sam se .NET znatno razvijao te uključuje različite komponente koje uvelike olakšavaju razvoj aplikacija. Na slici 2.4. prikazana je struktura odnosno sadržane komponente kroz različite verzije .NET Frameworka, kako je i prazano u [5].



Sl. 2.4. *Komponente Microsoft .NET Frameworka* [5]

Između ostalog, za izradu aplikacije korištena je tehnologija dinamički generiranih web stranica (eng. *Active Server Pages*, **ASP.NET**). *ASP.NET Framework* podrazumijeva razvojnu platformu otvorenog koda za web aplikacije, temeljenu na .NET tehnologiji, razvijenu od strane Microsofta te namijenjenu stvaranju dinamičnih web stranica, aplikacija i usluga. Kao glavno sredstvo razvoja istog, ASP.NET koristi Web Forme (*.aspx* ekstenzija datoteke) sa statičnim HTML označavanjem, *code-behind* modelom te direktivama za izgradnju kako sučelja tako i sadržaja. Pored ASP.NET tehnologije, u svrhu povezivanja s bazom podataka korištena je i tehnologija *ActiveX* podatkovnih objekata (eng. *ActiveX Data Objects*, **ADO.NET**) tehnologija – Microsoftova .NET tehnologija pristupa podacima koja omogućuje komunikaciju s relacijskim ili nerelacijskim bazama podataka.

Entity Framework predstavlja razvojnu platformu otvorenog koda objektno-relacijskog mapiranja, kao komponentu ADO.NET odnosno, općenito, .NET tehnologije. Entity Framework omogućuje razvoj podatkovno-orijentiranih aplikacija, gdje se uz više razine apstrakcije stvara virtualna objektno-orijentirana baza podataka kao način konvertiranja podataka između neodgovarajućih tipova sustava u objektno-orijentiranim jezicima. U ovom završnom radu korištena je inačica Entity Framework 5.0 koja je još uvijek u sklopu .NET tehnologije.

Hipertekstualni označni jezik (eng. *HyperText Markup Language*, **HTML**) – standardni je označni jezik (eng. *markup language* – jezik za obilježavanje i sintaktičko razlikovanje dokumenta od teksta) za stvaranje web stranica i web aplikacija. Na osnovu takvih HTML dokumenata koje dobiva od servera, web pretraživač iste pretvara u multimedijske web stranice kojim korisnici

pristupaju. Tipično obilježje HTML-a je upotreba oznaka (*tags*) u obliku kutnih zagrada (</>), kojima se uvodi sadržaj u web stranice te isti pobliže opisuju. Izgled tipičnog HTML koda prikazan je slikom 2.5. Zajedno sa tehnologijom kaskadnih listi oblikovanja (eng. *Cascading Style Sheets*, **CSS**) te **JavaScript** tehnologijom, HTML čini okosnicu WWW (eng. *World Wide Web*) razvoja i dizajna. Uz pomoć JavaScripta, HTML integrira programe vezane za ponašanje samih web stranice, dok je CSS-om definiran izgled i način prikaza sadržaja.

Koristeći sve spomenute programe, alate i tehnologije pristupilo se konkretnom rješavanju zadatka završnog rada – ponajprije dizajniranje i realizacija potrebne baze podataka uz pomoć Microsoft SQL Servera, zatim kreiranje aplikacije u Microsoft Visual Studiju, povezivanje s bazom podataka uporabom Entity Frameworka te daljnjim razvojem i pisanjem koda za prikaz, odabir i izmjenu baze podataka te konačno generiranje testa s pitanjima. Postupak je opisan u idućim poglavljima.

3. BAZA PODATAKA

Okosnica pravilnog funkcioniranja ove aplikacije te njene učinkovitosti jest upravo ispravno osmišljena te na adekvatan način realizirana baza podataka koja će pružiti pohranu svih potrebnih informacija (pitanja, parametara, testova i sl.) nužnih za rad aplikacije te generiranje testa.

Za izradu takve baze podataka ponajprije je osmišljen ER dijagram koji prikazuje entitete i veze među njima. Na temelju njega, izrađen je relacijski model odnosno tablice i entiteti sa svojim atributima i međusobnim odnosima. Na osnovu relacijskog modela, baza je realizirana uz pomoć SQL upita, pri čemu su pojedine tablice kreirane u bazi podataka te popunjene testnim primjerima odnosno podacima.

3.1. ER Dijagram

Za realizaciju dijagrama entitetskih odnosa (eng. *Entity Relationship,* **ER**), potrebni su sljedeći **entiteti**:

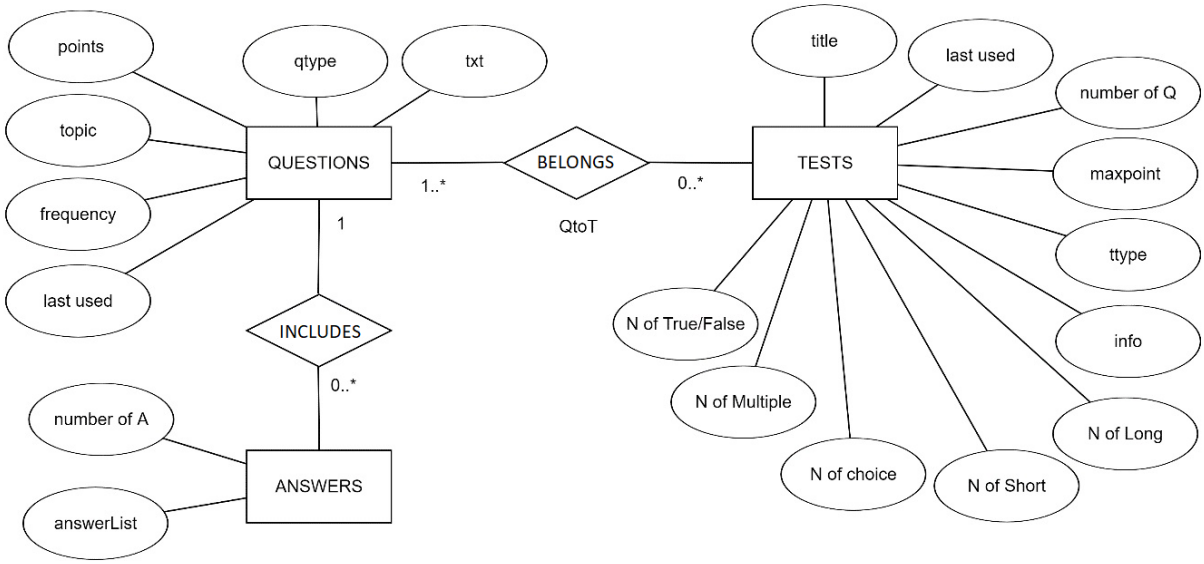
- QUESTIONS (skup svih pitanja)
- TESTS (skup svih testova)
- ANSWERS (skup svih ponuđenih odgovora koji idu uz neka pitanja,

te **veze** BELONGS i INCLUDES. Slika 3.1. prikazuje ER dijagram koji je pobliže opisan u nastavku.

Pitanje (QUESTIONS) pripada jednom ili više testova, ali može i ne pripadati nijednom testu (npr. nije se našlo niti u jednom testu dosad). Test (TESTS) sadrži jedno ili više pitanja od kojih je sačinjen. Važno je napomenuti kako se u ovom slučaju radi o vezi *više-više* što će rezultirati posebnom tablicom pri prelasku u relacijski model. Nadalje, za neka pitanja – kao što su pitanja odabira točnog odgovora ili višestrukog odabira – postoje odgovori u obliku liste ponuđenih odgovora. Pitanja (QUESTIONS) mogu sadržavati jedan ili više ponuđenih odgovora ili pak nijedan odgovor (ako se ne radi o spomenutim vrstama pitanja). S druge strane, ponuđeni odgovori (ANSWERS) pripadaju točno jednom određenom pitanju na koje se odnose.

Uz to, entiteti posjeduju određene **atribute** koji ih pobliže opisuju i omogućuje pravilno funkcioniranje. Tako je za pitanja (QUESTIONS) bitna vrsta pitanja (*qtype*), tekst pitanja (*txt*), broj bodova koje nosi pitanje (*points*), tema odnosno predmet (*topic*), učestalost pojavljivanja

(*frequency*) te vrijeme zadnje uporabe pitanja (*last used*). Slično i za testove (TESTS) bitan je naziv testa, vrijeme zadnje uporabe, ukupan broj pitanja, broj pitanja pojedine vrste pitanja, maksimalan broj bodova, vrsta testa i dodatne informacije o testu. Za odgovore (ANSWERS) potrebno je poznavati broj odgovora te listu ponuđenih odgovora.



Sl. 3.1. *ER dijagram*

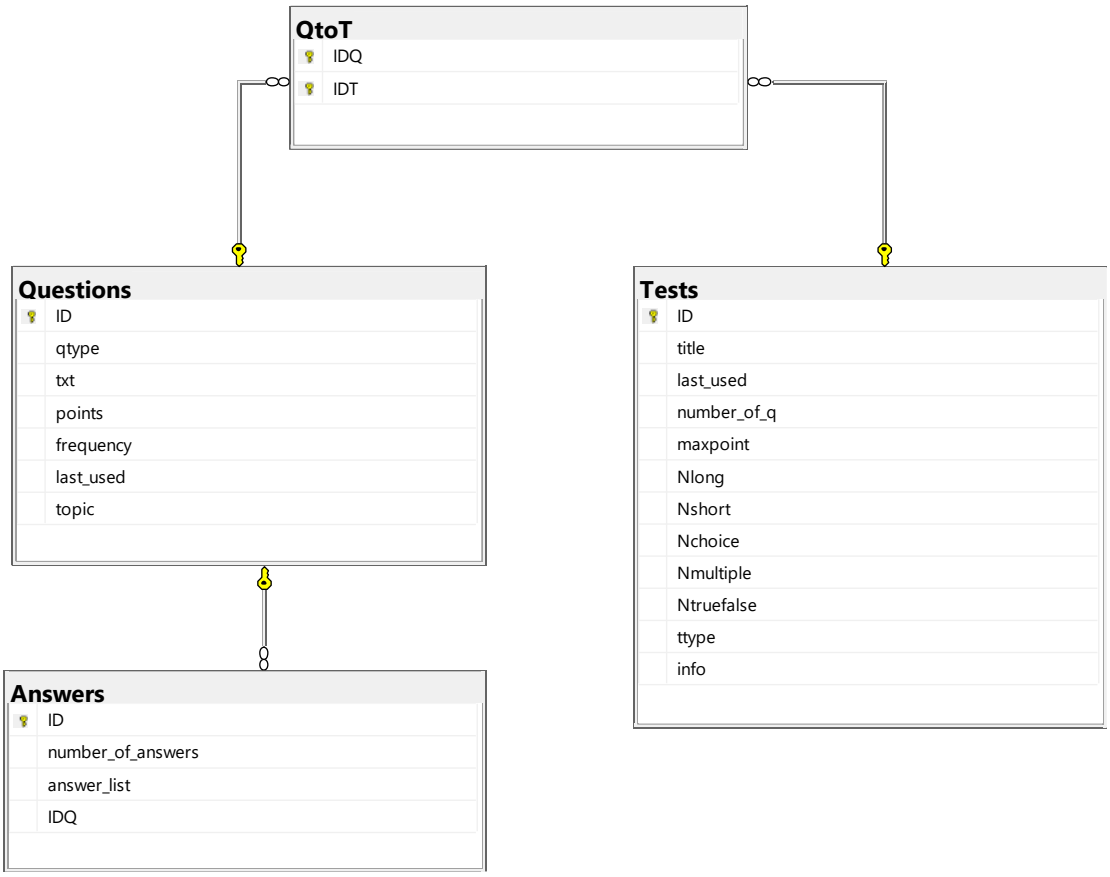
Na osnovu prikazanog i opisanog ER dijagrama, pristupa se izradi relacijskom modela podataka, na temelju čega je moguće realizirati tablice i konkretnu bazu SQL upitima.

3.2. Relacijski model

Koristeći spomenute entitete, veze između njih i atribute, ER dijagram se prevodi u relacijski model pogodan za realizaciju tablicama. Svaki od entiteta prelazi u zasebnu **relaciju** (tablicu) poštujući pravila prelaska iz ER dijagrama u relacijski model te nastaje sljedeća relacijska shema:

QUESTIONS (**ID**, qtype, txt, points, frequency, last_used, topic)
TESTS (**ID**, title, last_used, number_of_q, maxpoint, Nlong, Nshort, Nchoice, Nmultiple, Ntruefalse, ttype, info)
ANSWERS (**ID**, number_of_answers, answer_list, IDQ)
QtoT (**IDQ**, **IDT**)

Za svaku relaciju (tablicu) dodan je primarni ključ koji je prikazan podebljanim i podvučenim slovima. Strani ključevi prikazani su podvučenim slovima. Pri prelasku u relacijski model, na temelju veze BELONGS, koja je bila brojnosti *više-više*, pojavila se nova relacija (tablica) QtoT (*Questions-to-Tests*), koja ima složeni primarni ključ sastavljen od stranih ključeva koji se referenciraju na primarne ključeve pitanja i testova. Na temelju takve relacijske sheme, stvoren je relacijski model koji je grafički prikazan sljedećim dijagramom.



Sl. 3.2. *ER dijagram*

Preostaje isti relacijski model realizirati SQL upitima odnosno kreirati tablice baze podataka.

3.3. Realizacija SQL upitima

Kao što je prethodno spomenuto, za realizaciju baze podataka korišten je Microsoft SQL Server Management Studio. Ponajprije je stvorena nova baza podataka (*QUIZ_BUILDER*), u njoj novi upit (*query*) u kojem su sljedećim SQL upitima (Sl. 3.3) kreirane potrebne tablice iz relacijskog modela.

```
CREATE TABLE Questions (
ID INT PRIMARY KEY IDENTITY(1,1),
qtype VARCHAR(20),
txt TEXT,
points TINYINT DEFAULT 1,
frequency INT DEFAULT 0,
last_used SMALLDATETIME DEFAULT NULL ,
topic VARCHAR(50) DEFAULT NULL
);

CREATE TABLE Answers(
ID INT PRIMARY KEY IDENTITY(1,1),
number_of_answers TINYINT,
answer_list TEXT NOT NULL,
IDQ INT FOREIGN KEY REFERENCES Questions(ID)
);

CREATE TABLE Tests(
ID INT PRIMARY KEY IDENTITY(1,1),
title VARCHAR (30),
last_used SMALLDATETIME,
number_of_q TINYINT NOT NULL,
maxpoint TINYINT NOT NULL,
Nlong TINYINT,
Nshort TINYINT,
Nchoice TINYINT,
Nmultiple TINYINT,
Ntruefalse TINYINT,
ttype VARCHAR (20),
info TEXT
);


CREATE TABLE QtoT(
IDQ INT,
IDT INT,
CONSTRAINT pk_qtot PRIMARY KEY (IDQ, IDT),
CONSTRAINT fk_q FOREIGN KEY (IDQ) REFERENCES Questions (ID),
CONSTRAINT fk_t FOREIGN KEY (IDT) REFERENCES Tests (ID)
);
```

Sl. 3.3. *SQL kod za kreiranje tablica*

Kreirana baza odnosno tablice popunjene su testnim primjerima, tj. podacima (pitanja, odgovori, testovi, povezivanje odgovora i testova) u svrhu lakšeg rada i testiranja baze i aplikacije tijekom izrade. Sljedeći korak je izrada same web aplikacije i povezivanje sa stvorenom bazom.

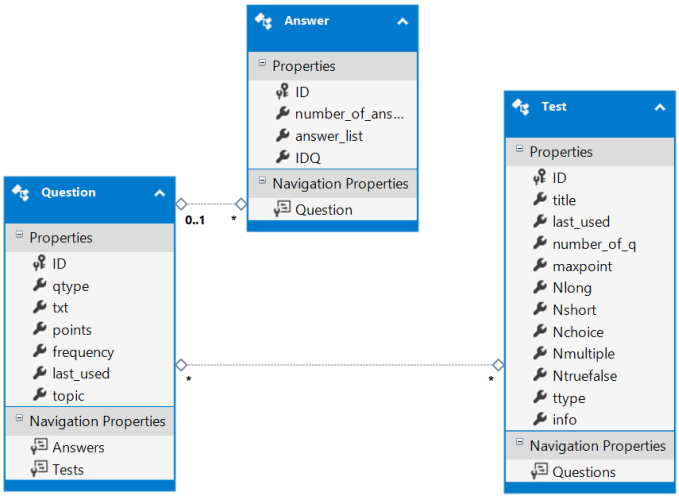
4. APLIKACIJA

Kao što je prethodno spomenuto, zadatak je riješen kao ASP.NET web aplikacija (C# programski jezik) u Visual Studio razvojnom okruženju. Kreiran je projekt pod nazivom *Generator_testova* kao jedan od ponuđenih predložaka ASP.NET web aplikacije s već kreiranom postavom stranice (*Site.master* datoteka). Ponajprije će se objasniti način povezivanja kreiranog projekta s prethodno stvorenom bazom podataka, zatim će se opisati pojedine ključne datoteke nužne za realizaciju aplikacije, uz objašnjenje bitnijih dijelova koda te na koncu i potrebne dodatne funkcionalnosti za svrsishodan rad aplikacije.

4.1. Povezivanje s bazom

Za povezivanje s bazom, kao što je spomenuto u prethodnim poglavljima (2.3. i 3.3.), korišten je Entity Framework. U Visual Studiju, uz otvoren projekt *Generator_testova* potrebno je među *NuGet Package* proširenjima, odabrati i instalirati ***Entity Framework*** za ovaj projekt. U ovom slučaju je korištena inačica *Entity Framework 5.0*. Nakon instaliranja, u *Solution Exploreru* Visual Studia, dodaje se nova datoteka za povezivanje s bazom. Pod predlošcima odabire se *Data > ADO.NET Entity Data Model* te se nova datoteka naziva *GeneratorDBModel.*

Zatim se kroz niz prozora redom odabire generiranje na osnovu postojeće baze podataka (*EF Designer from database*), stvara se i uspostavlja veza (*connection*) sa serverom te odabire način autorizacije. Po uspostavljenoj i testiranoj vezi sa serverom, postavke iste spremaju se u datoteku projekta *Web.config* pod nazivom *QUIZ_BUILDERDBContext* (datoteka objašnjena u idućim poglavljima). Tada se program spaja na server, prikuplja i prikazuje podatke te se odabiru objekti iz baze. Nakon odabranih tablica i odabranog naziva modela postupak povezivanja je gotov i u projekt je dodana datoteka *GeneratorDBModel.edmx* koja sadržava datoteke potrebne za realizaciju modela podataka te, između ostalog, i dijagram modela podataka (prikazan na slici 4.1.).



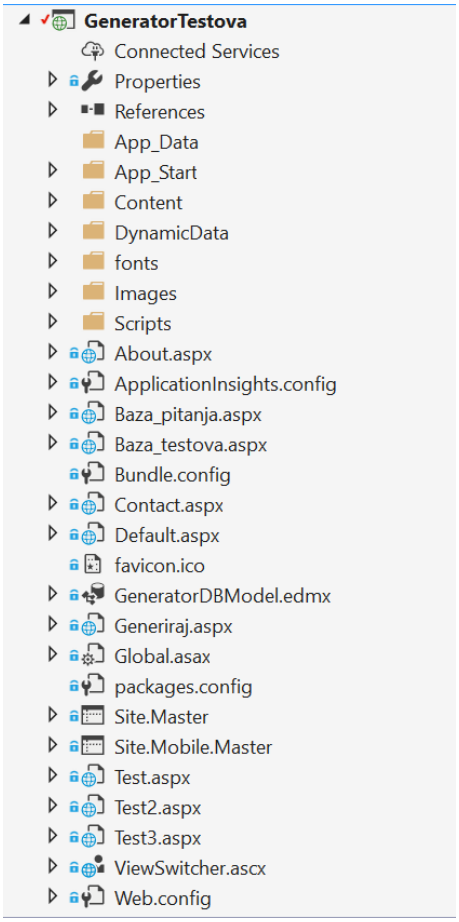
Sl. 4.1. *Model podataka aplikacije*

Na osnovu stvorenog modela podataka, uz pomoć Entity Frameworka automatski su generirane i potrebne klase koje će aplikacija koristiti. Sada je aplikacija povezana sa serverom i bazom podataka.

4.2. Ključne datoteke

Pri otvaranju samog projekta, u *Solution Exploreru* (Sl 4.2.), daju se uočiti bitne datoteke za aplikaciju. Pri tome su za dizajn i razvoj ovog projekta prvenstveno bitne datoteke za izgled sučelja, pojedinih web formi i funkcionalnosti, povezivanja sa serverom i slično.

Za većinu datoteka, naročito web formi, radi se i u *Source* prikazu i u *Design* prikazu. U *Source* prikazu na raspolaganju je cijeli označni kod koji definira web stranicu, dok je izgled iste odnosno raspored elemenata vidljiv u *Design* prikazu. Glavni dio datoteka su upravo web forme (*.aspx* ekstenzija) čija se izrada može promatrati u tri aspekta – *Design* koji prikazuje elemente i grubi izgled sučelja, *Source* koji prikazuje označni kod web stranice te *Code-behind* kojim se uređuje C# kod pojedinih elemenata forme (*.cs* ekstenzije datoteka sadržanih pod datotekama web formi). Dakle, za pojedine web forme, bitan je ponajprije označni HTLM kod koji definira elemente stranice, njihov izgled i raspored te C# *code-behind* kod kojim se definiraju određene funkcije, metode i varijable kako bi se odgovarajuće akcije mogle izvršavati.



Sl. 4.2. *Datoteke projekta*

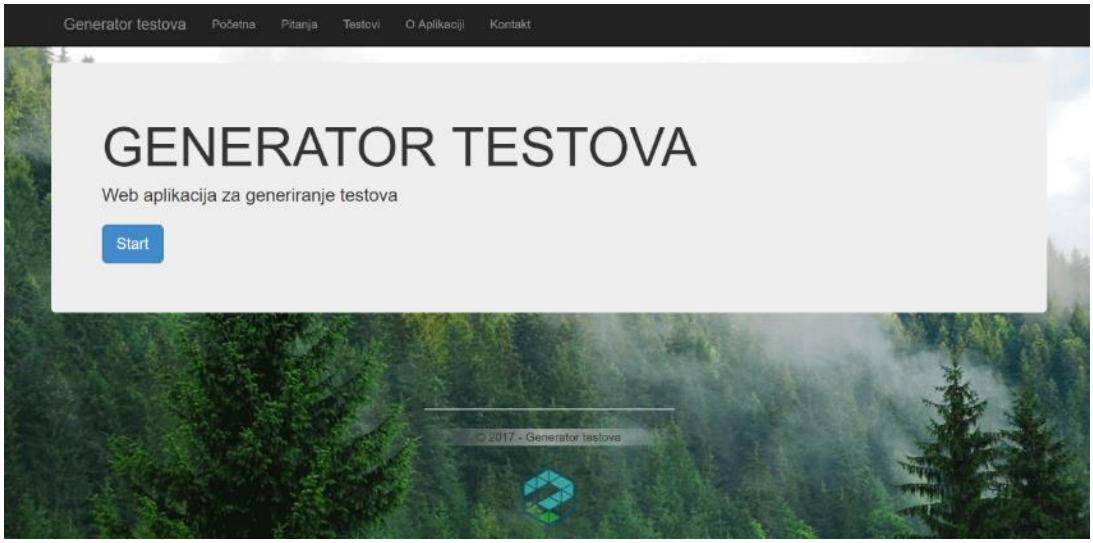
Prva bitna datoteka od koje počinje dizajn aplikacije je već spomenuta ***Site.master*** datoteka u kojoj se definira opći izgled web stranice koji preuzimaju ostale web forme. Podešava se izgled zaglavlja, linkovi na ostale web forme, boje sučelja i slično. Na samom početku definirane su potrebne skripte (biblioteke i funkcije) koje će se upotrebljavati u aplikaciji (npr. ispis testa ili otvaranje u novoj kartici pretraživača). Nadalje, označnim kodom su na glavnoj traci web stranice dodani linkovi za početnu stranicu, generator testova, stranicu o aplikaciji, kontaktu te baze pitanja i testova.

Pored toga, u skladu s dodanim linkovima, kreirane su web forme koje odgovaraju istim. Tako je za početnu stranicu kreirana web forma odnosno datoteka *Default.aspx*, za stranicu o aplikaciji *About.aspx*, za stranicu o kontaktu *Contact.aspx*, za stranicu za generiranje nasumičnog testa *Generiraj.aspx*, za stranicu s bazom pitanja i osobni odabir istih za test *Baza_pitanja.aspx*, stranice za prikaz i pregled testova *Test.aspx*, *Test2.aspx* i *Test3.aspx* te stranica s bazom testova *Baza_testova.aspx*. Slika 4.3. prikazuje dio koda kojim se dodani linkovi povezuju sa spomenutim web formama.

```
<div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" runat="server" href="~/Generiraj.aspx">Generator testova</a>
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li><a runat="server" href="~/">Početna</a></li>
        <li><a runat="server" href="~/Baza_pitanja">Pitanja</a></li>
        <li><a runat="server" href="~/Baza_testova">Testovi</a></li>
        <li><a runat="server" href="~/About">O Aplikaciji</a></li>
        <li><a runat="server" href="~/Contact">Kontakt</a></li>
```

Sl. 4.3. *Povezivanje linkova Site.Master datoteke s web formama*

Datoteka ***Default.aspx***, čiji je izgled prikazan na slici 4.4., predstavlja stranicu koja se prva prikazuje pri pokretanju aplikacije te je ujedno i početna stranica. Na njoj je pored općeg teksta o samoj aplikaciji dodan i gumb za odlazak na stranicu za generiranje testa (web forma *Generiraj.aspx*).



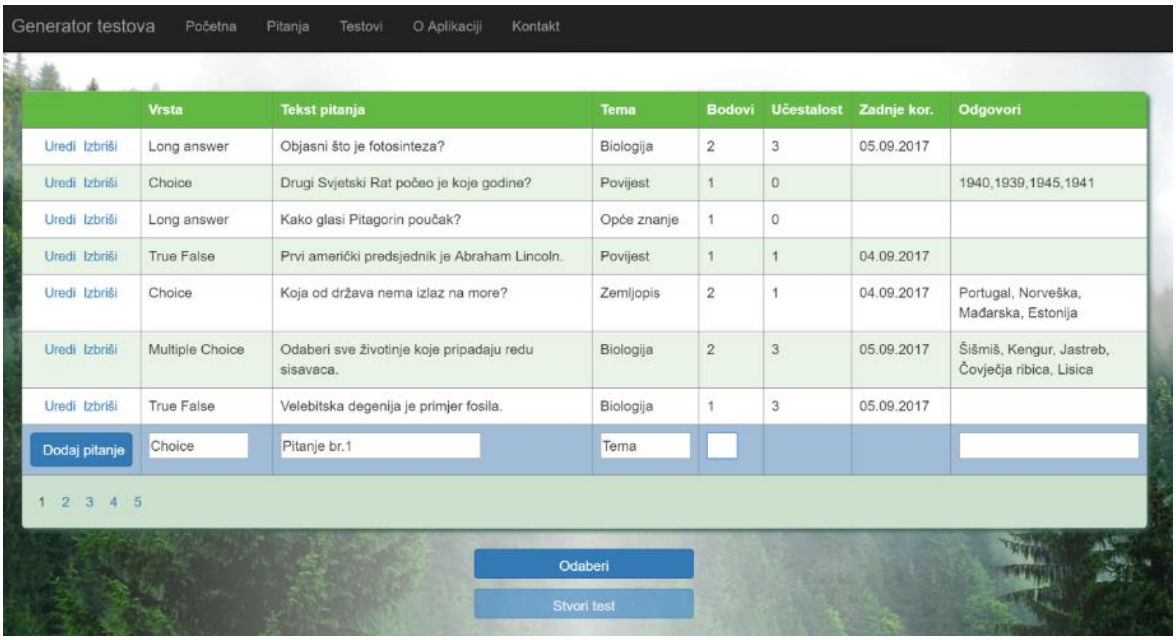
Sl. 4.4. *Izgled početne stranice*

Datoteka ***About.aspx*** namijenjena je pružanju više informacija o samoj aplikaciji i načinu rada, dok ***Contacts.aspx*** pruža kontakt informacije o autoru iste. Obje web forme su jednostavno kreirane s običnim dodavanjem teksta na sučelje.

4.2.1. Datoteka *Baza_pitanja.aspx*

Spomenuta web forma ***Baza_pitanja.aspx*** nosi dvije glavne funkcije – prikazuje bazu svih pitanja uz mogućnost dodavanja, brisanja i uređivanja istih, što je ostvareno povezivanjem sa serverom odnosno kreiranom bazom podataka, ali istovremeno može poslužiti i kao stranica za kreiranje testa s osobnim odabirom pitanja. Sama web forma sadrži *GridView 'Qgv'* za tablični prikaz podataka iz baze te *EntityDataSource 'BazaDataSource'* kojim se odabire izvor podataka za prikaz. Uz to sadrži pomoćne elemente za kreiranje testa osobnim odabirom pitanja čije su vidljivosti onemogućene: tablicu (*GridView) 'Qselected'* u koju se spremaju odabrana pitanja, tablicu (*GridView) 'Acorresponding'* iz koje se dohvaćaju odgovori za određena pitanja, *Label 'SadrzajTesta'* u koju se sprema cjelokupan tekst koji će predstavljati sadržaj testa. Tu su i *Label 'Notification'* za prikaz obavijesti pri određenim akcijama na stranici, dio stranice za unos parametara (ostvaren uz pomoć raznih *TextBoxova),* dio za ispis napomena pri unosima te gumbi (*Button)* za pokretanje odabira pitanja, stvaranje testa, prikaz i spremanje istog u bazu te odlazak na generiranje nasumičnog testa.

Glavni dio stranice predstavlja spomenuta tablica *'Qgv'* (Sl 4.5), koja sadrži više polja za prikaz i uređivanje pitanja te pri dnu i polja za unos novog pitanja na stisak odgovarajućeg gumba.



Sl. 4.5. *Glavna tablica Baze pitanja*

Kako bi tablica funkcionirala, ponajprije se u *EntityDataSource* podešava izvor podataka – odabiremo naziv veze odnosno prethodno imenovanu konekciju *QUIZ_BUILDERDBContext* te se zatim odabire željeni entitet (tablica *Questions*). Tada se u postavkama *GridView*-a odabire, kao

izvor podataka, prethodno postavljeni *EntityDataSource* te podešava željeni izgled tabličnog prikaza podataka (stupci za prikaz, nazivi, boje i sl.). Uz to, u tablici se prikazuju i odgovori, za pitanja koja imaju odgovore. Isto je postignuto uz pomoć svojstva *Include* u spomenutom izvoru podataka *EntityDataSource*, gdje je dodana tablica *Answers* iz baze podataka pa se podatci o odgovorima spremaju u dodani *ListView* odgovarajućeg *TemplateFielda* (Sl. 4.6.).

```
    <asp:GridView ID="Qgv" Width="1126" CssClass="table rounded-corners" RowStyle-VerticalAlign="Middle"
                  runat="server" DataSourceID="BazaDataSource" PageSize="7" ShowFooter="True"
                  AutoGenerateColumns="False" DataKeyNames="ID" AllowPaging="True" AllowSorting="True"
                  OnRowDeleting="Delete" OnRowDataBound="OnRowDataBound"
                  OnRowEditing="Qgv_RowEditing" OnRowUpdating="Update" OnPageIndexChanging="Qgv_PageIndexChanging">
    <Columns>...</Columns>
        <HeaderStyle BackColor="#63b843" ForeColor="White"/>
        <PagerStyle BackColor="#cde0cd" Height="10%"/>
        <RowStyle VerticalAlign="Middle" BackColor="#ffffff" />
        <FooterStyle BackColor ="#a0bcd6" />
        <AlternatingRowStyle BackColor ="#eaf4e6"
            />
</asp:GridView>
```

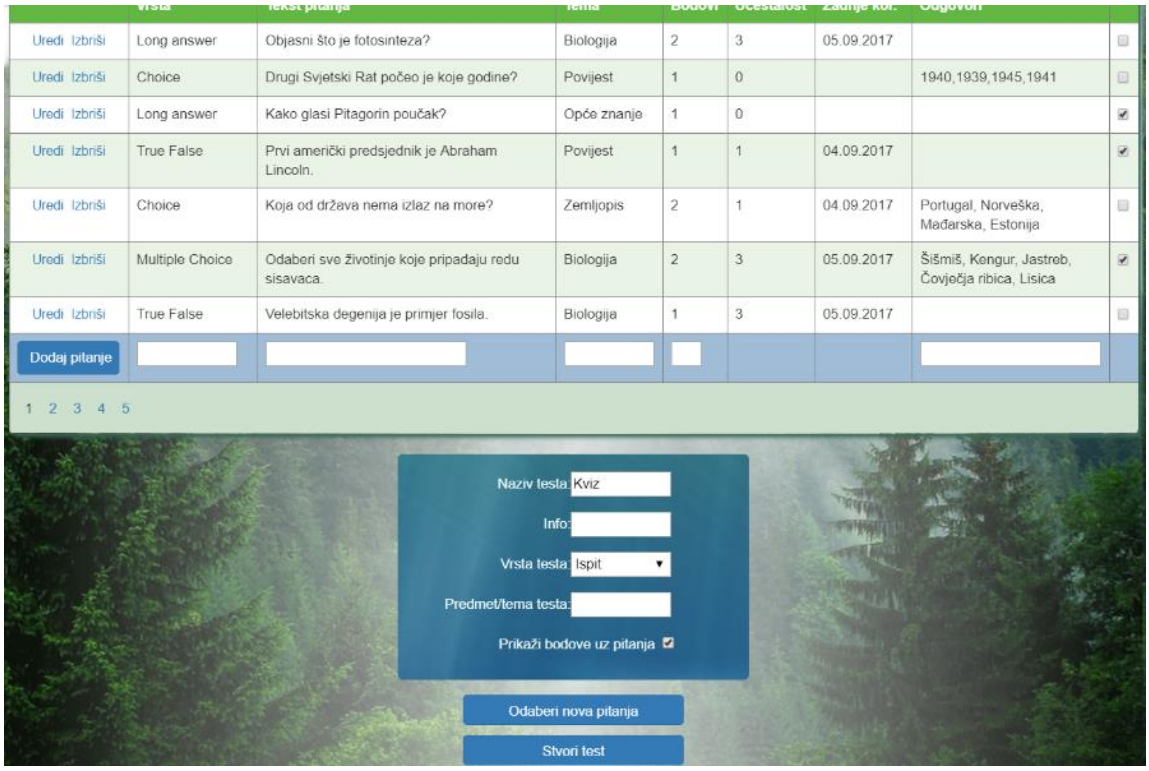
```
<asp:TemplateField HeaderText="Odgovori">
    <EditItemTemplate>
        <asp:ListView ID="ListView2" runat="server" DataSource='<%# Eval("Answers") %>' >
            <ItemTemplate>
            <table ><tr ><td style="border-style:none; padding:0px;">
                    <asp:Textbox Height="25px" id="OdgovoriUpdate" runat="server" Text='<%# Eval("answer_list") %>'>
                    </asp:Textbox>
            </td></tr></table>
            </ItemTemplate>
        </asp:ListView>
    </EditItemTemplate>

    <ItemTemplate>
        <asp:ListView ID="ListView1" runat="server" DataSource='<%# Eval("Answers") %>' >
            <ItemTemplate>
            <table ><tr ><td style="border-style:none; padding:0px;">
                    <asp:Label BorderStyle="None" Height="15px" id="ans" runat="server" Text='<%# Eval("answer_list") %>'>
                    </asp:Label>
            </td></tr></table>
            </ItemTemplate>
        </asp:ListView>
    </ItemTemplate>

    <FooterTemplate>
        <asp:TextBox ID="OdgovoriNew" Width = "180px" runat="server"></asp:TextBox>
    </FooterTemplate>
</asp:TemplateField>
```

Sl. 4.6. *Označni kod tablice te dijela za odgovore (unutar Columns dijela tablice)*

U *Source* prikazu kodom ili u *Design* prikazu među postavkama *GridView*-a, omogućuju se opcije (gumbi) za uređivanje (*Edit*) te brisanje podataka (*Delete*) iz baze. Da bi iste funkcionirale, polja (stupci) za koja se žele omogućiti spomenute opcije, potrebno je konvertirati u *TemplateField* odlaskom na *GridView > Edit Columns > Convert this field into a TemplateField.* Nakon što su konvertirani, u *Source* prikazu se mogu dodatno podešavati označnim jezikom (formati, načini provjere i sl.). Pri tome se te automatske funkcije nadograđuju uz pomoć vlastitih funkcija (npr. za uređivanje i odgovora, brisanje pitanja i popratnih odgovora ili unos novih

pitanja). To je postignuto uz pomoć C# funkcija u *code-behind* dijelu stranice. Između ostalog, tu su funkcije za dohvat podataka i spremanje istih (*GetData* i *InsertData*), za unos novih pitanja (*Insert*), uređivanje (*Update*) i brisanje (*Delete*) pitanja s odgovorima i slično. Iste se pokreću na stisak odgovarajućih gumba ili na određenu akciju u tablici (promjena stranice u tablici – *Qgv_PageIndexChanging*, izmjena podataka – *Qgv_RowEditing* i sl.).

Druga bitna funkcija ove stranice, kao što je spomenuto, jest upravo **stvaranje testa na osnovu vlastitog odabira pitanja** (Sl 4.7). Pritiskom na gumb *'Odaberi'* u tablici se pojavljuju *Checkboxi* kojima se isto omogućuje te dio za unos parametara (naziv, tema i sl.). Nakon što su pitanja odabrana, pritiskom na gumb *'Stvori test'* pokreće se napisana C# funkcija '*Kreiraj_Click'* (prolazi se kroz cijelu tablicu *'Qgv'* i ako se naiđe na pitanje odnosno red kod koga je *CheckBox* označen, isto se sprema u pomoćnu tablicu '*Qselected'* na osnovu čega se oblikuje sadržaj testa – sam proces generiranja testa opisan je kod *Generiraj.aspx*) te se na osnovu odabranih pitanja stvara test koji je moguće pregledati i ispisati te spremiti u bazu testova.



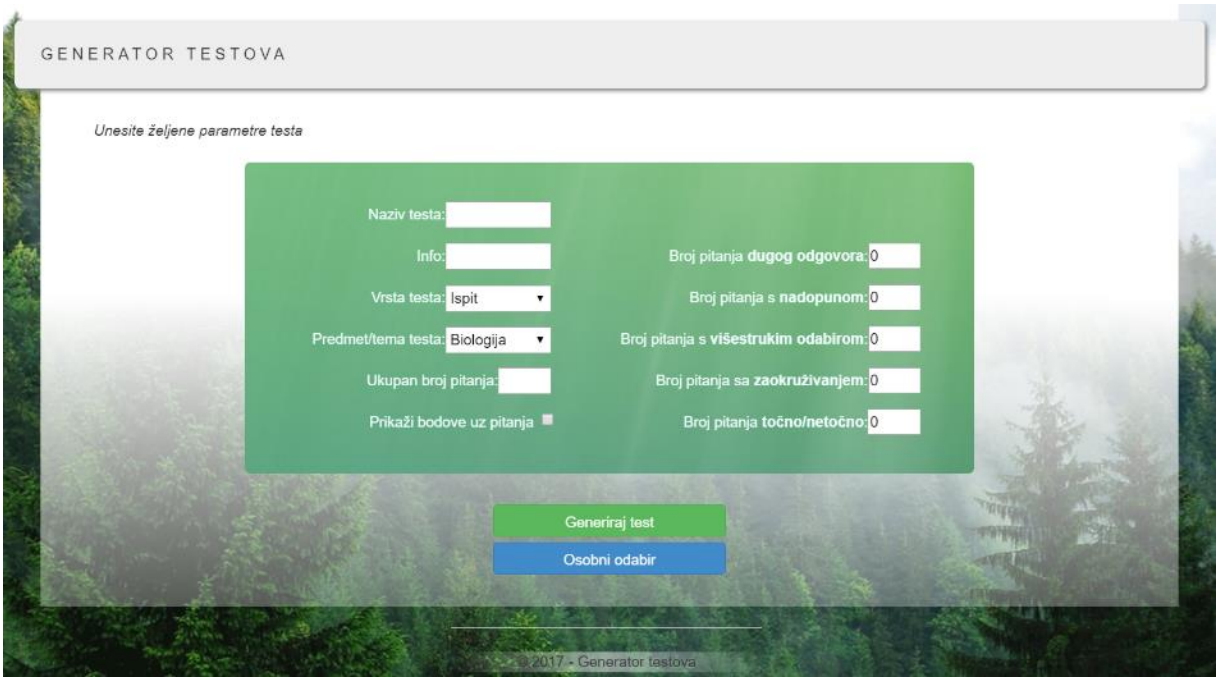
Sl. 4.7. *Osobni odabir pitanja za test*

Za navedeno se također koriste dodane C# funkcije (metode) kao što su *'Prikazi'* (za prikaz testa koristeći web formu ***Test2.aspx*** koja se uz pomoć JavaScript funkcije *SetTarget()* otvara u novoj kartici te se oslanja na *Baza_pitanja.aspx* kao prethodnu stranicu od koje dobiva potrebni

SadrzajTesta kako bi isti bio prikazan na stranici), *'SpremiUBazu'* (za spremanje testa u bazu testova koristeći pomoćne funkcije *GetData* i *InsertData* te stvoreni model podataka, pri čemu test postaje vidljiv u *Baza_testova.aspx*). Na stranici za prikaz testa *Test2.aspx* omogućen je gumb kojim se pokreće JavaScript funkcija *printpage()* koja otvara novi prozor te automatski pokreće ispis testa (ili spremanje istog u .pdf formatu na računalo). Obje spomenute JavaScript funkcije prethodno su definirane u zaglavlju *Site.Master* stranice.

4.2.2. Datoteka *Generiraj.aspx*

Web forma ***Generiraj.aspx*** omogućuje korisniku unos parametara željenog testa – za iste sadrži tekst koji opisuje pojedini parametar te *TextBox*: za upis naziva testa, dodatnih informacija o testu, za željeni ukupni broj pitanja te broj pitanja pojedine vrste pitanja (dugog odgovora, kratkog odgovora/nadopune, višestrukog odabira, odabira točnog odgovora te odabira točno/netočno), teme testa te opcije prikaza bodova pitanja. Uz to, uključen je i *DropDownList* element kojim se omogućuje odabir vrste testa. Na dnu se nalazi gumb (*button*) kojim se pokreće dio koda za generiranje testa. Slika 4.8. prikazuje izgled stranice za generiranje testa (*Generiraj.aspx*) u pregledniku.



Sl. 4.8. *Izgled stranice za generiranje testa (Generiraj.aspx)*

U svom označnom kodu, pored spomenutih elemenata, sadrži i skrivene pomoćne elemente (analogno web formi *Baza_pitanja.aspx*) za generiranje testa, a u C# kodu – funkcije kojima se

između ostalog izvršava generiranje testa na osnovu unesenih parametara, spremanje istog, dohvat i unos podataka, prikaz testa i sl. (Sl. 4.9).

```
0 references | Vedran, 4 hours ago | 1 author, 1 change | 0 exceptions
protected void GenerirajTest(object sender, EventArgs e)...

0 references | Vedran, 4 hours ago | 1 author, 1 change | 0 exceptions
protected void PrikaziTest(object sender, EventArgs e)...


0 references | Vedran, 4 hours ago | 1 author, 1 change | 0 exceptions
protected void SpremiUBazu(object sender, EventArgs e)...

//funkcija za dohvat podataka na osnovu zadanog queryja
7 references | Vedran, 4 hours ago | 1 author, 1 change | 0 exceptions
private static DataTable GetData(string query)...

//funkcija za unos podataka na osnovu zadanog queryja
1 reference | Vedran, 4 hours ago | 1 author, 1 change | 0 exceptions
public void InsertData(DataRow Row, string TableName)...
```

Sl. 4.9. *C# kod – neke od funkcija (Generiraj.aspx)*

Generiranje (funkcija *GenerirajTest*) funkcionira na sljedeći način: ponajprije se pomoćnom funkcijom *GetData* stvara tablica podataka na osnovu broja pojedinačnih pitanja, gdje se SQL upitom (Sl. 4.10.) dohvaća onoliko proizvoljnih pitanja odabrane teme za svaku vrstu koliko je korisnik zatražio (nasumičnost se osigurava uz pomoć *„ORDER BY newid()“* sintakse).

```
protected void GenerirajTest(object sender, EventArgs e)
{
    if (IsDigits(TUkupno.Text) && IsDigits(TLong.Text) && IsDigits(TShort.Text) && IsDigits(TMulti.Text)
        && IsDigits(TChoice.Text) && IsDigits(TTF.Text) && TNaziv.Text.Length != 0)
    {
        if (int.Parse(TUkupno.Text) == int.Parse(TLong.Text) + int.Parse(TShort.Text)
            + int.Parse(TMulti.Text) + int.Parse(TChoice.Text) + int.Parse(TTF.Text))
        {
            //Generiranje na osnovu parametara:
            DataTable Q = GetData("SELECT TOP " + TLong.Text + " * FROM Questions WHERE qtype = 'Long Answer' AND topic = '"
                + TTema.SelectedValue.ToString().ToLower() + "' ORDER BY newid();");
            Q.Merge(GetData("SELECT TOP " + TShort.Text + " * FROM Questions WHERE qtype = 'Short Answer' AND topic = '"
                + TTema.SelectedValue.ToString().ToLower() + "' ORDER BY newid();"));
            Q.Merge(GetData("SELECT TOP " + TChoice.Text + " * FROM Questions WHERE qtype = 'Choice' AND topic = '"
                + TTema.SelectedValue.ToString().ToLower() + "' ORDER BY newid();"));
            Q.Merge(GetData("SELECT TOP " + TMulti.Text + " * FROM Questions WHERE qtype = 'Multiple Choice' AND topic = '"
                + TTema.SelectedValue.ToString().ToLower() + "' ORDER BY newid();"));
            Q.Merge(GetData("SELECT TOP " + TTF.Text + " * FROM Questions WHERE qtype = 'True False' AND topic = '"
                + TTema.SelectedValue.ToString().ToLower() + "' ORDER BY newid();"));

            QGenerated.DataSource = Q;
            QGenerated.DataBind();

            ACorresponding.DataSource = GetData("Select * FROM Answers");
            ACorresponding.DataBind();
```

Sl. 4.10. *Proces generiranja testa (dohvat pitanja na osnovu parametara te popunjavanje dviju pomoćnih tablica)*

Tada se uz pomoć dvije dodatne skrivene tablice, ovisno o vrsti pitanja, formatira sadržaj testa (dodavanje „TOČNO“ i „NETOČNO“ za *True/False* tip pitanja ili npr. dodavanje nekoliko

linija za pitanja dugog odgovora), što je prikazano na slici 4.11.. Takav sadržaj se sprema u tekst skrivenog *Labela* za prikaz testa (Sl. 4.12.).

```
//Popunjavanje testa pitanjima
int i; String Popis = ""; MaxP.Text = "0";
for (i = 0; i < QGenerated.Rows.Count; i++)
{
    Popis = Popis + (i + 1).ToString() + ". " + QGenerated.Rows[i].Cells[2].Text + "<br />";
    if (QGenerated.Rows[i].Cells[1].Text == "True False")
    {
        Popis = Popis + "<pre style =\"...\" > TOČNO      NETOČNO </pre>";
    }
    if (QGenerated.Rows[i].Cells[1].Text.ToLower() == "long answer")
    {
        for (int k = 0; k < 3; k++)
        {
            Popis = Popis + "<div style =\"... border-bottom-style:solid; border-bottom-width:0.5px;\"> </div>";
        }
    }
    if (QGenerated.Rows[i].Cells[1].Text.ToLower() == "short answer")
    {
        Popis = Popis.Substring(0, Popis.Length - 6) + "<div style =\"... border-bottom-style:solid; border-bottom-width:0.5px;\"> </div>";
    }
    if (QGenerated.Rows[i].Cells[1].Text.ToLower() == "choice" || QGenerated.Rows[i].Cells[1].Text.ToLower() == "multiple choice")
    {
        char c = 'a';
        foreach (GridViewRow R in ACorresponding.Rows)
        {
            if (R.Cells[2].Text == QGenerated.Rows[i].Cells[0].Text)
            {
                string[] ans = R.Cells[1].Text.Split(',').Select(sValue => sValue.Trim()).ToArray();
                for (int k = 0; k < int.Parse(R.Cells[0].Text); k++)
                {
                    Popis = Popis + "<pre style =\"...\" > " + (char)(c + k) + ") " + ans[k] + "\n</pre>";
                }
                break;
            }
        }
    }
```

Sl. 4.11. *Proces generiranja testa (stvaranje i formatiranje sadržaja testa ovisno o vrsti pitanja)*

```
            // Prikaz bodova pojedinih pitanja ovisno o chekiranom parametru
            Popis = Popis + ((TBodovi.Checked == true) ? "<div style =\"...\">" + QGenerated.Rows[i].Cells[3].Text + "</div>" : "<br />");
            MaxP.Text = (int.Parse(MaxP.Text) + int.Parse(QGenerated.Rows[i].Cells[3].Text)).ToString();
        }
        Popis = Popis + "<div style =\"...\">/" + MaxP.Text.ToString() + "</div>";
        SadrzajTesta.Text = Popis;

        SpremiBaza.Enabled = true;
        Ispocetka.Enabled = true;
        Prikazi.Enabled = true;
        Prikazi.Visible = true;
        Ispocetka.Visible = true;
        SpremiBaza.Visible = true;
        Generiraj.Text = "Generiraj novi test";
        Notification.Text = "<span style=\"...\">Test je uspješno generiran<span>";
    }
    else Notification.Text = "Ukupan zbroj pojedinačnih pitanja se ne podudara s ukupnim brojem pitanja!";
}
else
{
    Notification.Text = "Potrebno je unijeti sve parametre!";
}
```

Sl. 4.12. *Proces generiranja testa (dodavanje bodova i spremanje sadržaja u Label koji se predaje Test-formi za prikaz)*

Povezana datoteka ***Test.aspx*** kao prethodnu stranicu ima upravo *Generiraj.aspx* te kao takvoj mogu joj se slati neki od parametara (sadržaj testa – pitanja, informacije i naslov). Analogno bazi pitanja i stranici *Test2.aspx* i ova stranica se uz pomoć JavaScript funkcija otvara u novoj

kartici te omogućuje ispis testa (na stisak gumba pokreće se spomenuta JavaScript funkcija *printpage()* koja započinje ispis sadržaja točno zadanog odjeljka stranice – *div* s *ID=PRINT*). U Test-formi (u ovom slučaju *Test.aspx*) u označnom kodu dodani su elementi potrebni za prikaz testa (*Labeli*), a u C# kodu (Sl. 4.13.) – pridodijeljen im je potrebni sadržaj koristeći *PreviousPage* svojstvo.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (PreviousPage != null)
    {
        Page.Title = PreviousPage.Title1;

        Label1.Text = PreviousPage.Title1;
        Label2.Text = PreviousPage.Info;
        Label3.Text = PreviousPage.Sadrzaj;

        HtmlForm f = (HtmlForm)Page.Master.FindControl("MasterForm");
        if (f != null)
        {
            f.Target = "_self";          // _parent, _blank, _self ili _top
        }
    }
    else
    {
        Label1.Text = "...> Potrebno unijeti parametre za generiranje testa! <span>";
        Panel1.Visible = false;
        PrintOrPDF.Visible = false;
        UnesiParametre.Visible = true;
    }
```

Sl. 4.13. *Prikaz testa uz prijenos parametara s prethodne stranice (Test.aspx i njena prethodna stranica Generiraj.aspx)*

Tako generiran test, moguće je spremiti u bazu testova koristeći spomenutu funkciju *'SpremiUBazu'*. Slika 4.14. prikazuje način rada funkcije – najprije se stvara novi red tablice (pomoćna funkcija *GetData*) – test koji se popunjava podacima ovisno o parametrima te se unosi u bazu (pomoćna funkcija *InsertData*). Zatim se, koristeći stvoreni model podataka, međusobno povezuju test i pitanja koja su sadržana u njemu te se podaci za ta pitanja ažuriraju (učestalost i datum zadnje upotrebe).

```
protected void SpremiUBazu(object sender, EventArgs e)
{   //Stvaranje novog Testa za unos u bazu
    DataTable Tt = GetData("SELECT * FROM Tests;");
    DataRow Rt = Tt.NewRow();
    Rt["title"] = TNaziv.Text; Rt["last_used"] = DateTime.Now;
    Rt["number_of_q"] = QGenerated.Rows.Count; Rt["Nlong"] = byte.Parse(TLong.Text);
    Rt["Nshort"] = byte.Parse(TShort.Text); Rt["Nchoice"] = byte.Parse(TChoice.Text);
    Rt["Nmultiple"] = byte.Parse(TMulti.Text); Rt["Ntruefalse"] = byte.Parse(TTF.Text);
    Rt["maxpoint"] = byte.Parse(MaxP.Text); Rt["ttype"] = TVrsta.SelectedValue.ToString();
    Rt["ttopic"] = TTema.Text; Rt["info"] = TInfo.Text;
    InsertData(Rt, "Tests");
    Notification.Text = "Test je spremljen u bazu";
    Notification.ForeColor = System.Drawing.Color.Green;

    using (QUIZ_BUILDERContext entities = new QUIZ_BUILDERContext())
    {
        Test T = entities.Tests.OrderByDescending(t => t.ID).FirstOrDefault();
        foreach (GridViewRow R in QGenerated.Rows)
        {
            int ID = int.Parse(R.Cells[0].Text);
            Question Q = (from c in entities.Questions where c.ID == ID select c).FirstOrDefault();
            Q.Tests.Add(T);
            T.Questions.Add(Q);
            if (Q != null)
            {
                if (Q.frequency > 0) Q.frequency++;
                else Q.frequency = 1;
                Q.last_used = DateTime.Today;
                entities.SaveChanges();
            }
        }
    }
}
```

Sl. 4.14. Označni kod *Baze_testova*

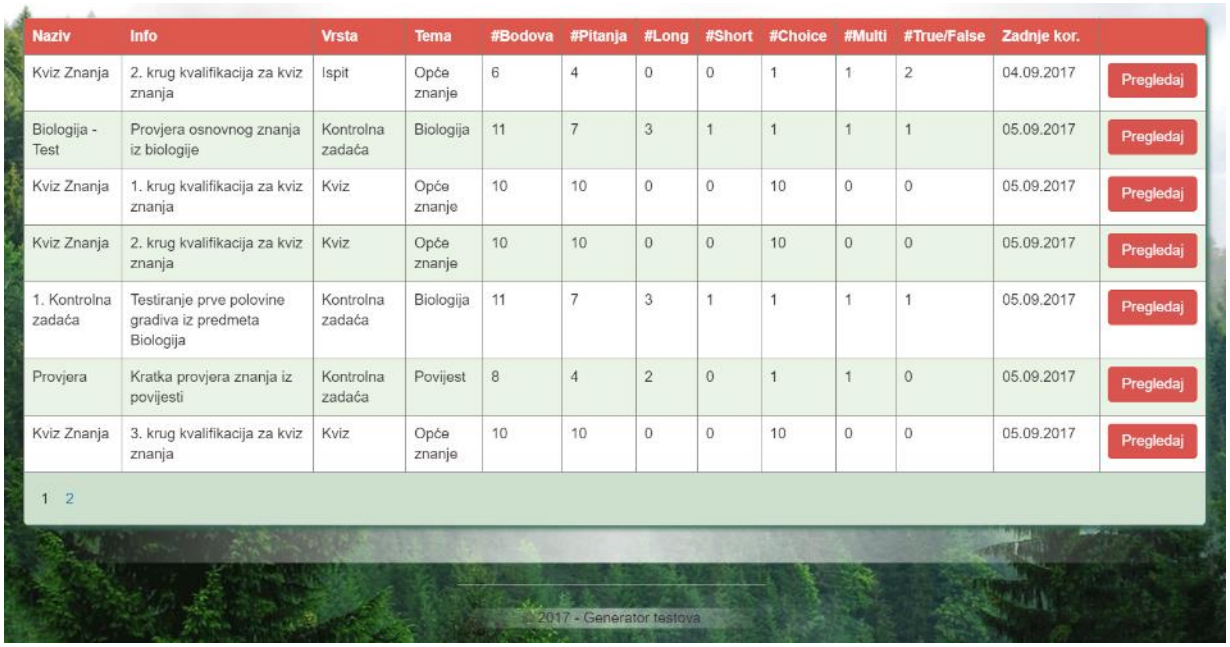
4.2.3. Datoteka *Baza_testova.aspx*

Posljednja ključna datoteka je web forma za prikaz baze testova. Slika 4.15. prikazuje njen označni kod (HTML elemente) među kojima se izdvaja: glavna tablica (*GridView*) *'Tgv'* za prikaz testova, *EntityDataSource* kao izvor podataka, dvije skrivene pomoćne tablice *'QUsed'* i *'ACorresponding'* te tri *Labela* za rekonstrukciju sadržaja testa. Tablica (*GridView*) kao glavni izvor podataka ima tablicu *Tests* baze podataka, ali uz svojstvo *Include* spomenutog izvora podataka uključuje i tablicu *Questions* pomoću koje se dohvaća ID svih pitanja sadržanih u pojedinom testu. Isto je postignuto na sličan način opisan kod *Baze_pitanja.aspx*, gdje se podatci iz druge tablice spremaju u *ListView* unutar dodanog *TemplateFielda*. Samo polje za prikaz liste ID-a upotrebljenih pitanja je skriveno od korisnika, ali se i dalje koristi za rekonstrukciju testa.

```
<center style="width: 1076px; padding:20px; padding-bottom:15px;
              background:linear-gradient(to right, rgba(255, 255, 255, 0), rgba(255,255,255,0.8),rgba(255,255,255,0));
              position:fixed center; margin-left:35px; ">
    <div style="display:block; margin-top:20px;">
          <asp:UpdatePanel ID="UpdatePanel1" runat="server">
              <ContentTemplate>
              <asp:GridView ID="Tgv" Width="1126px" runat="server" DataKeyNames="ID"
                     CssClass="table rounded-corners" RowStyle-VerticalAlign="Middle" PageSize="7"
                     AllowPaging="True" AllowSorting="True" AutoGenerateColumns="False"
                     DataSourceID="EntityDataSource1">
                     <Columns>...</Columns>
                     <HeaderStyle BackColor="#de584e" ForeColor="White"/> <PagerStyle BackColor="#cde0cd" Height="10%"/>
                     <RowStyle VerticalAlign="Middle" BackColor="#ffffff" /> <FooterStyle BackColor ="#a0bcd6" />
                     <AlternatingRowStyle BackColor ="#eaf4e6"/>
              </asp:GridView>
              </ContentTemplate>
                 <Triggers>...</Triggers>
          </asp:UpdatePanel>

          <asp:EntityDataSource ID="EntityDataSource1" runat="server"
           ConnectionString="name=QUIZ_BUILDERContext" DefaultContainerName="QUIZ_BUILDERContext"
           EnableFlattening="False" EntitySetName="Tests" Include="Questions" OrderBy="it.[last_used]">
          </asp:EntityDataSource>

          <asp:GridView ID="QUsed" runat="server" Visible="False"></asp:GridView>
          <asp:GridView ID="ACorresponding" ...>...</asp:GridView>
          <asp:Label ID="SadrzajTesta" runat="server" Visible="false"></asp:Label>
          <asp:Label ID="LNaziv" runat="server" Text="Label" Visible="False"></asp:Label>
          <asp:Label ID="LInfo" runat="server" Text="Label" Visible="False"></asp:Label>
    </div>
</center>
<div style="z-index:-100; background-image:url('Images/BP.png'); background-repeat:repeat-x;
            position:fixed; top:50px; left: 0px; width: 5500px; height:1234px; margin-right: 0px; margin-bottom: 0px;">
</div>
```

Sl. 4.15. Označni kod *Baze_testova*

Kako bi korisnik mogao ponovno pregledati željeni test – u tablicu je dodano polje za gumb *'Pregledaj'* (Sl. 4.16.) čijim se stiskom pokreće funkcija u C# kodu stranice *'Prikaz_Click'*. Navedena funkcija (prikazana na slici 4.17.) za odabrani test (onaj pored kojeg je pritisnut gumb) dohvaća ID-jeve pitanja koje je taj test sadržavao, na osnovu njih puni tablicu *'QUsed'* te uz pomoć druge dodatne tablice *'ACorresponding'* i dodatnih *Labela* na način opisan u *Generiraj.aspx* ponovno stvara željeni test (uz identično formatiranje ovisno o vrsti pitanja opisano u funkciji *GenerirajTest* kod spomenute stranice) te istovremeno otvara web formu ***Test3.aspx*** u kojoj ga i prikazuje.



Naziv	Info	Vrsta	Tema	#Bodova	#Pitanja	#Long	#Short	#Choice	#Multi	#True/False	Zadnje kor.	
Kviz Znanja	2. krug kvalifikacija za kviz znanja	Ispit	Opće znanje	6	4	0	0	1	1	2	04.09.2017	Pregledaj
Biologija - Test	Provjera osnovnog znanja iz biologije	Kontrolna zadaća	Biologija	11	7	3	1	1	1	1	05.09.2017	Pregledaj
Kviz Znanja	1. krug kvalifikacija za kviz znanja	Kviz	Opće znanje	10	10	0	0	10	0	0	05.09.2017	Pregledaj
Kviz Znanja	2. krug kvalifikacija za kviz znanja	Kviz	Opće znanje	10	10	0	0	10	0	0	05.09.2017	Pregledaj
1. Kontrolna zadaća	Testiranje prve polovine gradiva iz predmeta Biologija	Kontrolna zadaća	Biologija	11	7	3	1	1	1	1	05.09.2017	Pregledaj
Provjera	Kratka provjera znanja iz povijesti	Kontrolna zadaća	Povijest	8	4	2	0	1	1	0	05.09.2017	Pregledaj
Kviz Znanja	3. krug kvalifikacija za kviz znanja	Kviz	Opće znanje	10	10	0	0	10	0	0	05.09.2017	Pregledaj

1 2

2017 - Generator testova

Sl. 4.16. *Izgled tablice testova te gumbi za prikaz istih*

```
protected void Prikaz_Click(object sender, EventArgs e)
{
    DataTable Q = GetData("SELECT * FROM Questions WHERE ID = null");
    LinkButton btn = (LinkButton)sender;
    GridViewRow row = (GridViewRow)btn.NamingContainer;

    foreach (ListViewItem I in (row.Cells[13].FindControl("Pitanja") as ListView).Items)
    {
        Q.Merge(GetData("SELECT * FROM Questions WHERE ID = " + (I.FindControl("IDs") as Label).Text));
    }
    QUsed.DataSource = Q;
    QUsed.DataBind();
    ACorresponding.DataSource = GetData("Select * FROM Answers");
    ACorresponding.DataBind();

    int i; String Popis = ""; int MaxP = 0;
    for (i = 0; i < QUsed.Rows.Count; i++)
    {
        Popis = Popis + (i + 1).ToString() + ". " + QUsed.Rows[i].Cells[2].Text + "<br />";
        if (QUsed.Rows[i].Cells[1].Text == "True False")...

        if (QUsed.Rows[i].Cells[1].Text.ToLower() == "long answer")...

        if (QUsed.Rows[i].Cells[1].Text.ToLower() == "short answer")...

        if (QUsed.Rows[i].Cells[1].Text.ToLower() == "choice" || QUsed.Rows[i].Cells[1].Text.ToLower() == "multiple choice")...
        MaxP = MaxP + int.Parse(QUsed.Rows[i].Cells[3].Text);
        Popis = Popis + "<br />";
    }
    Popis = Popis + "<div style =\"...\">/" + MaxP.ToString() + "</div>";
    SadrzajTesta.Text = Popis;
    LNaziv.Text = row.Cells[1].Text;
    LInfo.Text = row.Cells[2].Text;
    Server.Transfer("~/Test3.aspx");
}
```

Sl. 4.17. *Funkcija Prikazi_Click za rekonstrukciju i prikaz testa*

4.3. Uređivanje stranica i ostale datoteke

Kako je jedan od zahtjeva aplikacije i samog zadatka završnog rada grafički pristupačno sučelje i dizajn, spomenute web forme potrebno je i urediti, odnosno stilizirati i dizajnirati elemente koje sadržavaju. Za navedeno korišten je CSS jezik unutar označnog koda koji opisuje kako HTML elementi izgledaju na stranici. Korištena je prethodno stvorena CSS biblioteka – datoteka ***bootstrap.css*** koja sadrži opsežan broj definiranih izgleda i klasa za oblikovanje elemenata. Između ostalog korištene su njene klase za izgled tablice (*table*), izgled gumba (*btn btn-primary*, *btn-success* te *btn-danger*) te skrivanje dijelova tablice (*hidden*).

Osim toga, korišteni su i vlastiti pisani **CSS kodovi** bilo za pojedinačnu stranicu (na početku unutar <style> oznaka) ili za pojedinačni element iste (unutar *tag-a* koristeći ključnu riječ ***style***). Tako je npr. za svaku stranicu dodana pozadina kao element ( ) koji je oblikovan CSS kodom (Sl. 4.18.) kako bi prikazivao odabranu sliku.

```
<div style="z-index:-100; background-image:url('Images/BP.png'); background-repeat:repeat-x;
            position:fixed; top:50px; left: 0px; width: 5500px; height:1234px; margin-right: 0px;
            margin-bottom: 0px;">
</div>
```

Sl. 4.18. *Korištenje 'inline' CSS koda*

Ako se radi o **CSS klasi** koja se želi dodijeliti određenom elementu stranice, unutar *tag-a* tog elementa se ubacuje *'CssClass="naziv_klase"'* za ASP.NET elemente ili općenito '*class="naziv_klase"'*. Slika 4.19. prikazuje primjer dodjeljivanja klase jedne tablice i jednog gumba.

```
<asp:GridView ID="Tgv" Width="1126px" runat="server" DataKeyNames="ID" CssClass="table rounded-corners"
    RowStyle-VerticalAlign="Middle" PageSize="7" AllowPaging="True" AllowSorting="True"
    AutoGenerateColumns="False" DataSourceID="EntityDataSource1">
    <Columns>...</Columns>
```

```
<asp:Button CssClass="btn btn-primary" ID="Odabir" Height="30px" Width="220px"
            runat="server" OnClick="Odaberi" OnClientClick="ResetTarget();" Text="Odaberi"/>
```

Sl. 4.19. *Pridjeljivanje CSS klase elementima*

Jednom pridijeljena CSS klasa mora biti i definirana – bilo unutar datoteke (.css) projekta ili na početku same stranice koja sadrži element. Slikom 4.20. prikazana je za primjer definicija klase *table.*

```css
table {
    max-width: 100%;
    background-color: transparent;
    margin-right: 0px;
    border-radius: 4px;
}

th {
  text-align: left;
}

.table {
    width: 100%;
    margin-bottom: 20px;
    border-radius: 4px;
}
```

Sl. 4.20. *Primjer definiranja CSS klase*

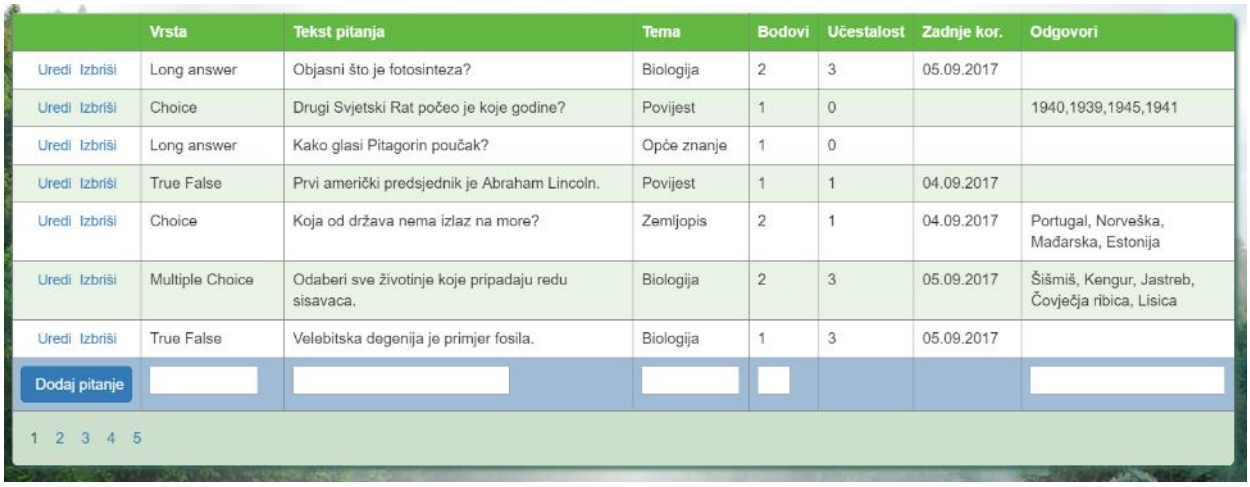
Uz pomoć CSS-a moguće je dakle odrediti **položaj elementa** (npr. svojstva: *margin, position, padding*), **izgled elementa** (npr. *background, background-image, background-color, border-style, width, height, box-shadow, display),* oblikovati **sadržani tekst** (npr. *font-style, font-weight, color, text-align*) te utjecati i na **elemente sadržane unutar** tog elementa (npr. *horizontal-align, vertical-align, line height*).

Za oblikovanje elemenata stranica aplikacije korištena su spomenuta te mnoga druga svojstva, a na slikama 4.21. i 4.22. se može uočiti jasna razlika između elementa (tablice) prije (Sl. 4.21) i nakon (Sl. 4.22) uređivanja CSS kodom. Na sličan način oblikovani su svi elementi web aplikacije.

	Vrsta	Tekst pitanja	Tema	Bodovi	Učestalost	Zadnje kor.	Odgovori
Uredi Izbriši	Long answer	Objasni što je fotosinteza?	Biologija	2	3	05.09.2017	
Uredi Izbriši	Choice	Drugi Svjetski Rat počeo je koje godine?	Povijest	1	0		1940,1939,1945,1941
Uredi Izbriši	Long answer	Kako glasi Pitagorin poučak?	Opće znanje	1	0		
Uredi Izbriši	True False	Prvi američki predsjednik je Abraham Lincoln.	Povijest	1	1	04.09.2017	
Uredi Izbriši	Choice	Koja od država nema izlaz na more?	Zemljopis	2	1	04.09.2017	Portugal, Norveška, Mađar Estonija
Uredi Izbriši	Multiple Choice	Odaberi sve životinje koje pripadaju redu sisavaca.	Biologija	2	3	05.09.2017	Šišmiš, Kengur, Jastreb, Č ribica, Lisica
Uredi Izbriši	True False	Velebitska degenija je primjer fosila.	Biologija	1	3	05.09.2017	
Dodaj pitanje							

12345

Sl. 4.21. *Tablica prije primjene CSS koda*



	Vrsta	Tekst pitanja	Tema	Bodovi	Učestalost	Zadnje kor.	Odgovori
Uredi Izbriši	Long answer	Objasni što je fotosinteza?	Biologija	2	3	05.09.2017	
Uredi Izbriši	Choice	Drugi Svjetski Rat počeo je koje godine?	Povijest	1	0		1940,1939,1945,1941
Uredi Izbriši	Long answer	Kako glasi Pitagorin poučak?	Opće znanje	1	0		
Uredi Izbriši	True False	Prvi američki predsjednik je Abraham Lincoln.	Povijest	1	1	04.09.2017	
Uredi Izbriši	Choice	Koja od država nema izlaz na more?	Zemljopis	2	1	04.09.2017	Portugal, Norveška, Mađarska, Estonija
Uredi Izbriši	Multiple Choice	Odaberi sve životinje koje pripadaju redu sisavaca.	Biologija	2	3	05.09.2017	Šišmiš, Kengur, Jastreb, Čovječja ribica, Lisica
Uredi Izbriši	True False	Velebitska degenija je primjer fosila.	Biologija	1	3	05.09.2017	
Dodaj pitanje							

1 2 3 4 5

Sl. 4.22. *Tablica nakon primjene CSS koda*

Sam projekt, web aplikacija, kako je spomenuto, organizirana je iz više datoteka i mapa. Između ostalog, bitno je naglasiti i mape: ***Content*** (koja sadrži .css datoteke), ***Images*** (koja sadrži korištene slike u aplikaciji), ***Scripts*** (koja sadrži skripte – JavaScript datoteke) te ***References*** (koja sadrži sve potrebne biblioteke – .dll datoteke, između ostalog i onu za Entity Framework). Uz to, pored ključnih datoteka objašnjenih u prethodnom poglavlju, za pisanje i rad aplikacije važne su i datoteke: *Web.config* datoteka te *GeneratorDBModel.edmx* datoteka.

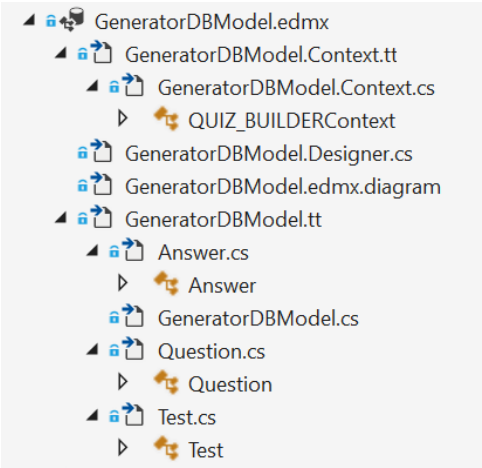
Web.config datoteka zadužena je za sva povezivanja aplikacije pa između ostalog sadrži i potrebne podatke o korištenom Entity Frameworku te vezi sa serverom. U projektu su korištena dva povezivanja sa serverom (Sl. 4.23.) – prvo je spomenuta veza *QUIZ_BUILDERContext* koja koristi Entity Framework za povezivanje, a drugo je vlastito kreirana veza *MyConn* koja koristi klasični *SqlClient* način povezivanja.

```
<connectionStrings>
    <add name="MyConn" connectionString="Server=VEKI-LENOVO; Database=QUIZ_BUILDER; User Id=korisnik; password=12345678"
         providerName="System.Data.SqlClient" />
  <add name="QUIZ_BUILDERContext" connectionString="metadata=res://*/GeneratorDBModel.csdl|res://*/GeneratorDBModel.ssdl|res://
           */GeneratorDBModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=.;
           initial catalog=QUIZ_BUILDER;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework
           &quot;" providerName="System.Data.EntityClient" />
</connectionStrings>
```

Sl. 4.23. *Web.Config datoteka (dio za povezivanja)*

GeneratorDBModel.edmx datoteka rezultat je povezivanja aplikacije s bazom podataka koristeći Entity Framework pri čemu je stvoren spomenuti model podataka. Isti podrazumijeva dijagram koji prikazuje odnose među entitetima te dodatne C# datoteke koje su automatski kreirane pri povezivanju, a sadržavaju klase kojima se definiraju entiteti (koji odgovaraju tablicama), njihovi atributi i međusobne veze. Slika 4.24. prikazuje datoteke potrebne za

realizaciju takvog modela podataka, a slika 4.25. primjer automatski generiranih klasa za entitet (tablicu iz povezane baze podataka).



Sl. 4.24. *Datoteke modela podataka*

```
public partial class Question
{
    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public Question()
    {
        this.Answers = new HashSet<Answer>();
        this.Tests = new HashSet<Test>();
    }

    3 references | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public int ID { get; set; }
    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public string qtype { get; set; }
    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public string txt { get; set; }
    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public Nullable<byte> points { get; set; }
    7 references | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public Nullable<int> frequency { get; set; }
    3 references | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public Nullable<System.DateTime> last_used { get; set; }
    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public string topic { get; set; }

    1 reference | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public virtual ICollection<Answer> Answers { get; set; }
    3 references | Vedran, 20 hours ago | 1 author, 1 change | 0 exceptions
    public virtual ICollection<Test> Tests { get; set; }
}
```

Sl. 4.25. *Klasa Question iz modela podataka*

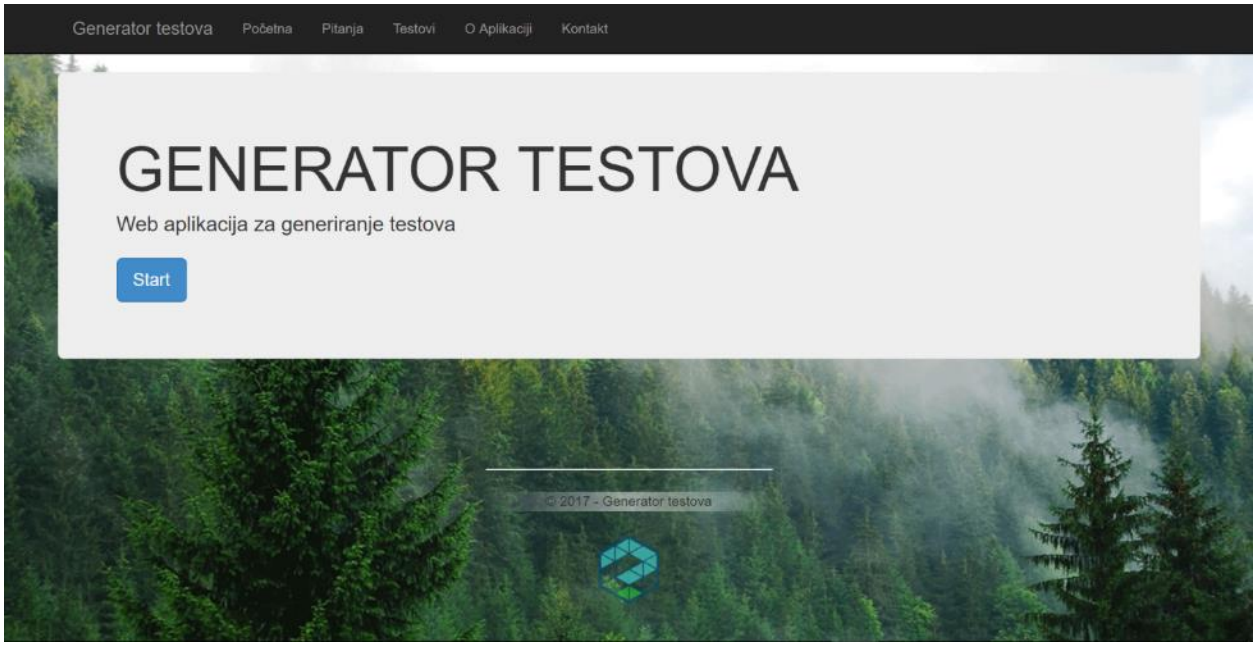
Time su opisani cjelokupni potrebi resursi i postupci za izradu potrebne web aplikacije. U idućem poglavlju, analiziran je i prikazan rad same aplikacije.

5. ANALIZA I VREDNOVANJE (*USER EXPERIENCE*)

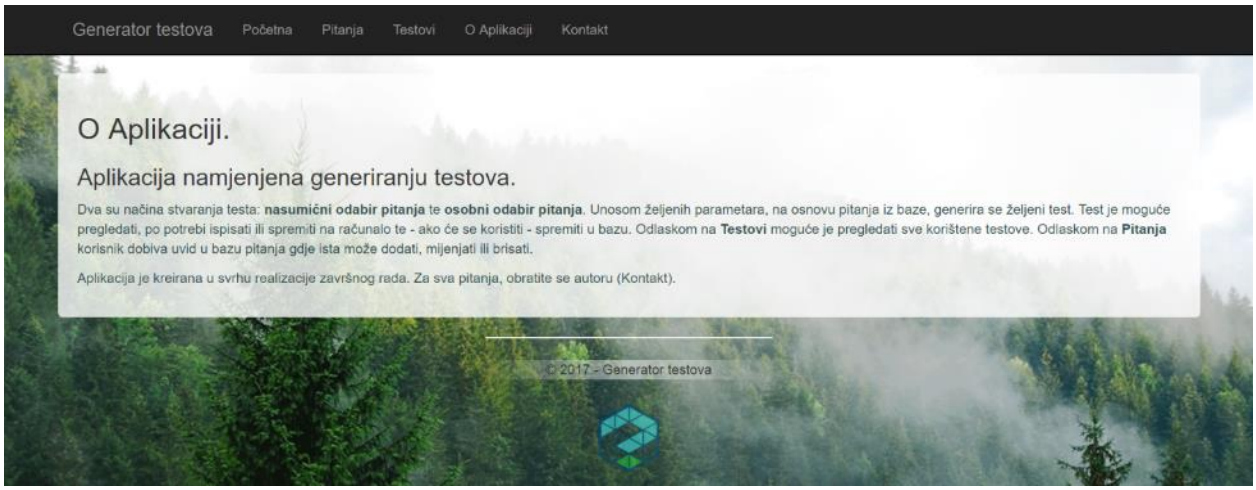
Po kreiranoj aplikaciji, pristupa se vrednovanju iste odnosno analizi njenog izgleda, funkcionalnosti, performansi, dizajna, brzine izvedbe i slično što je ukratko opisano u idućim poglavljima.

5.1. Izgled aplikacije

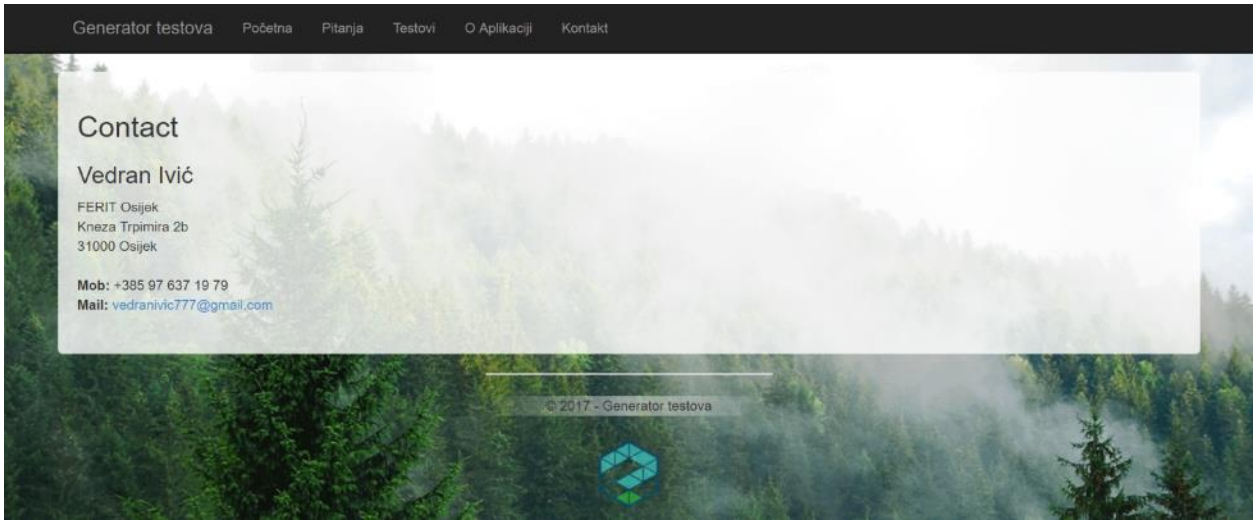
Aplikacija se kao ASP.NET web aplikacija temelji na web formama i gotovom osnovnom dizajnu predloška koji se da modificirati i prilagoditi po želji. Iz prethodnih poglavlja da se uočiti da postoji par glavnih sučelja aplikacije – **početna stranica**, stranica s podacima i informacijama **o aplikaciji**, njenom radu i slično, stranica **o kontaktu** (informacije o autoru), stranica za **prikaz baze pitanja i osobni odabir pitanja**, stranica za **generiranje nasumičnog testa**, stranica za **prikaz testa** te stranica za **prikaz baze testova**. Izgled pojedinih dan je sljedećim slikama (Sl. 5.1., Sl. 5.2., Sl. 5.3., Sl. 5.4., Sl. 5.5., Sl.5.6. i Sl.5.7.).



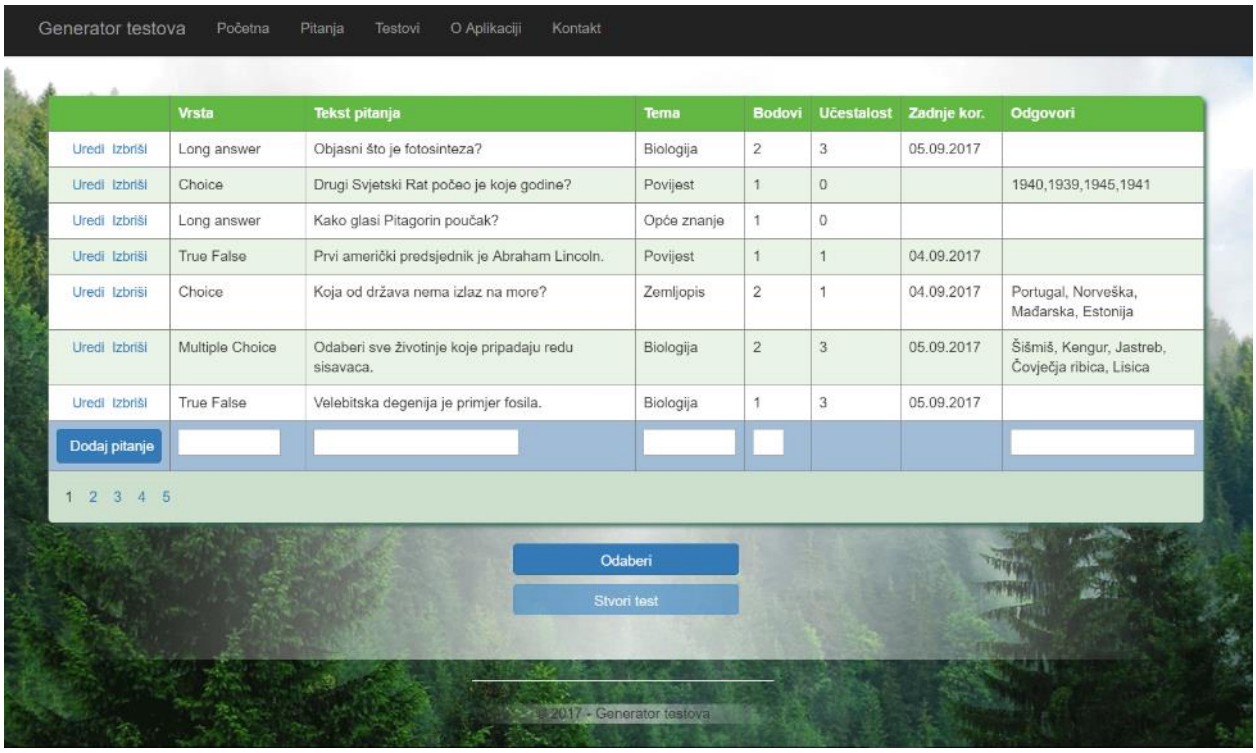
Sl. 5.1. *Početna stranica*



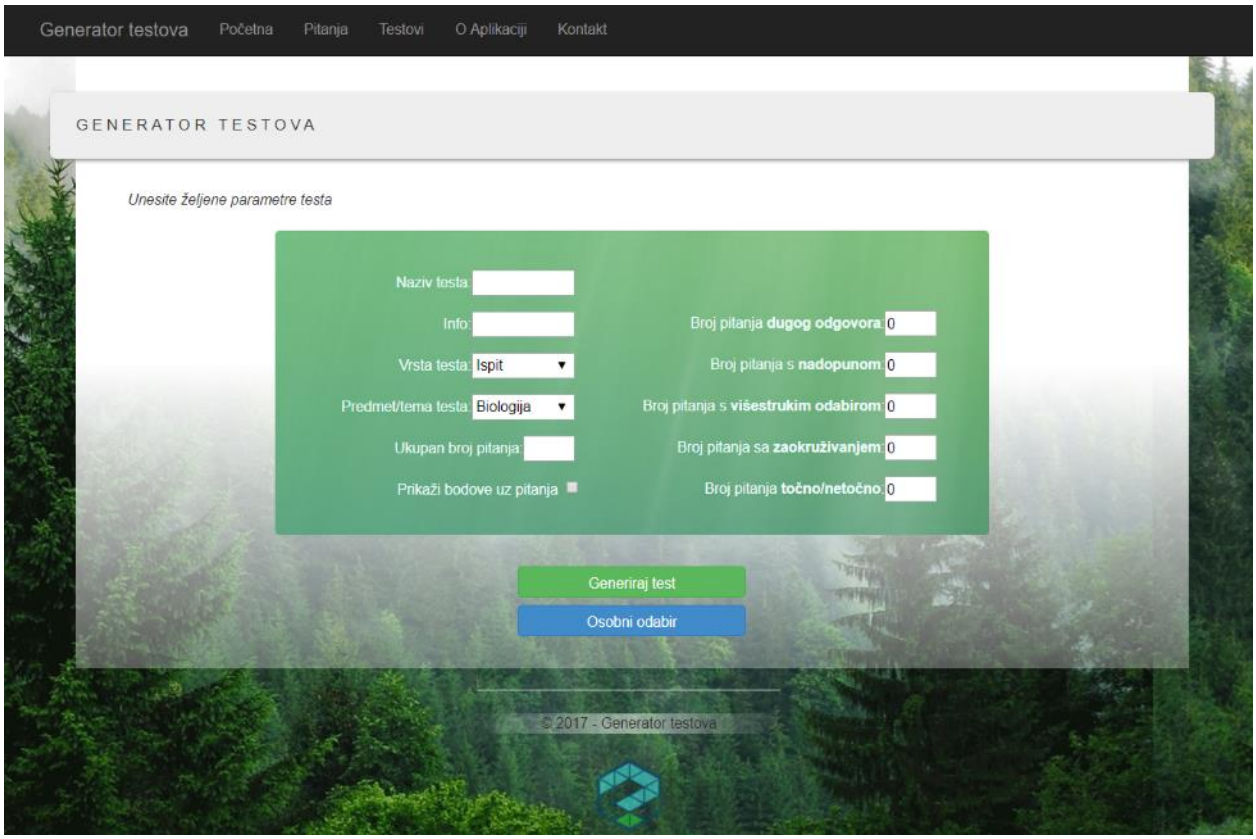
Sl. 5.2. *Stranica O Aplikaciji*



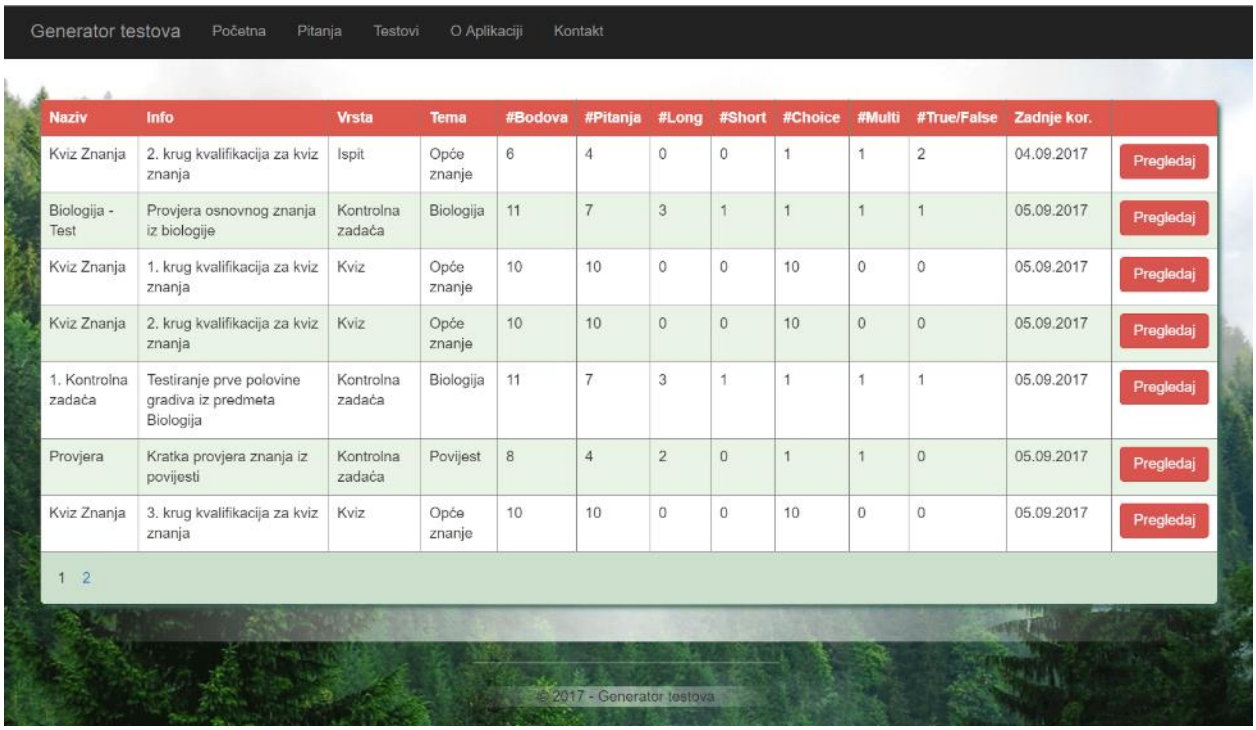
Sl. 5.3. *Stranica Kontakt*



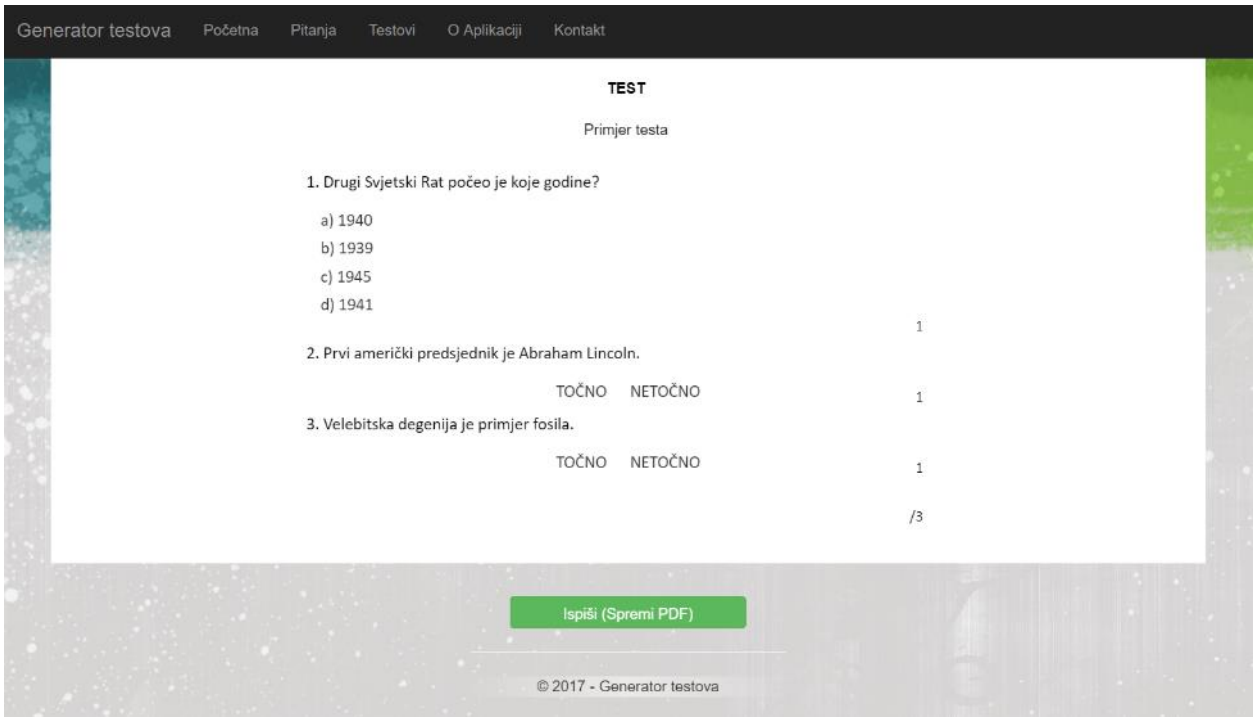
Sl. 5.4. *Stranica Pitanja*



Sl. 5.5. *Stranica Generator Testova*



Sl. 5.6. *Stranica Testovi*



Sl. 5.7. *Stranica za prikaz testa*

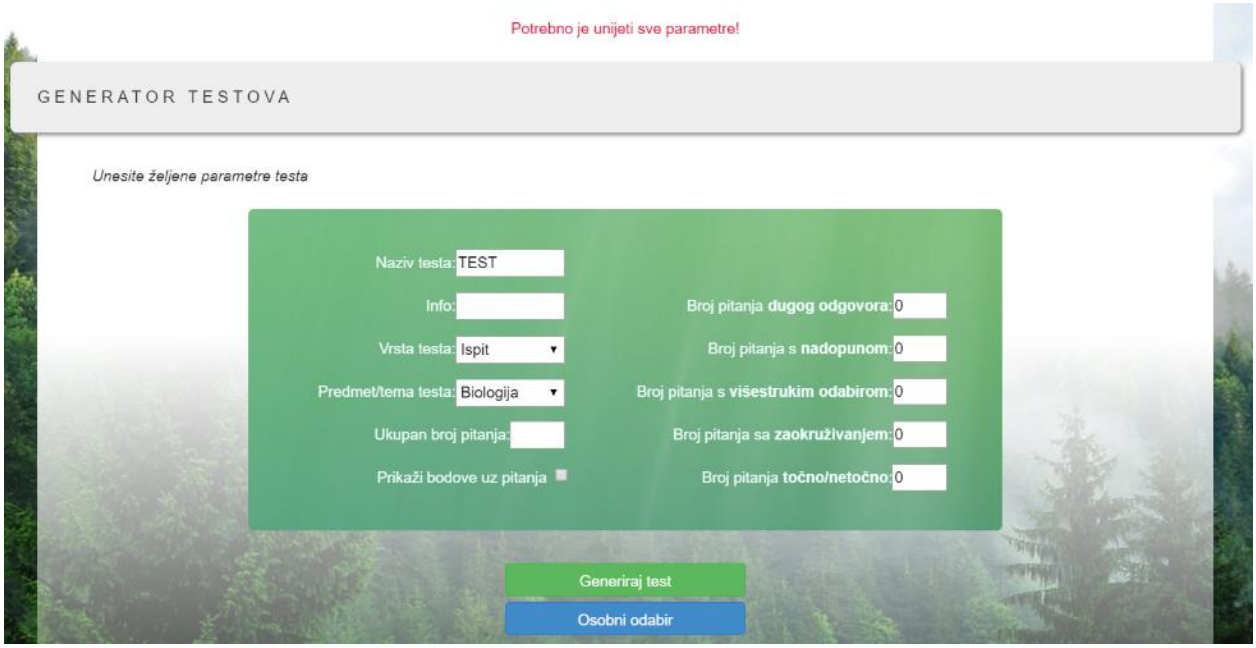
Aplikacija, dakle, pruža zadovoljavajuću razinu grafičkog sučelja, gdje je korištenje olakšano, intuitivno i jednostavno.

5.2. Testiranje

Za potrebe testiranja, baza podataka je popunjena s nekoliko pitanja različitih vrsta i tema, odgovarajućim odgovorima te nekoliko različitih testova. U svrhu analize rada aplikacije – testirat će se generiranje nasumičnog testa, generiranje testa s osobnim odabirom pitanja, spremanje testa u bazu i njegov ispis te rad s bazom pitanja i bazom testova.

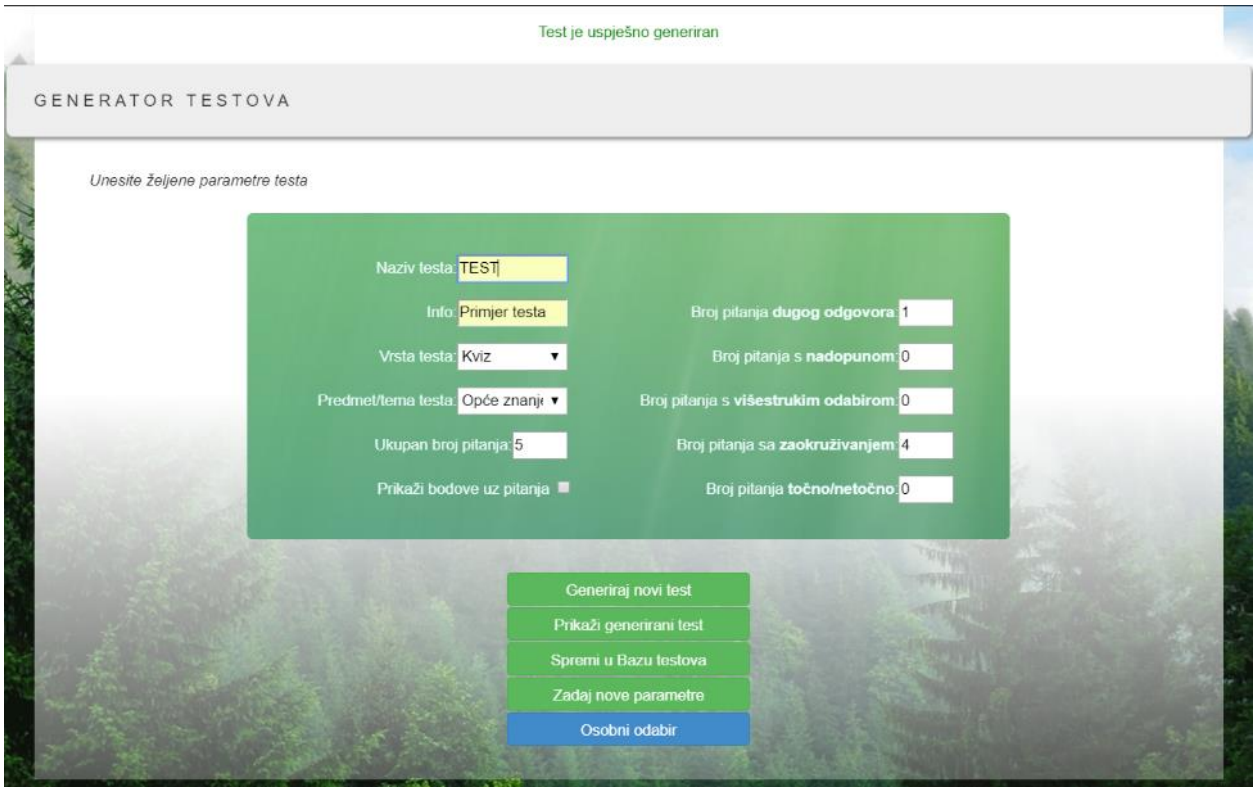
5.2.1. Generiranje nasumičnog testa

Pri pokretanju aplikacije, klikom na gumb *Start* ili na polje *Generator Testova* u izborničkoj traci, pristupa se stranici za nasumično generiranje testova. Na istoj se uočava kućica za unos parametara, gumb za generiranje te gumb za odlazak na osobni odabir. Za potrebe testiranja unose se probne vrijednosti. Potrebno je unijeti sve obavezne parametre – u suprotnom se ispisuje **poruka o greški** (Sl. 5.8.)



Sl. 5.8. *Poruka o neunesenim parametrima*

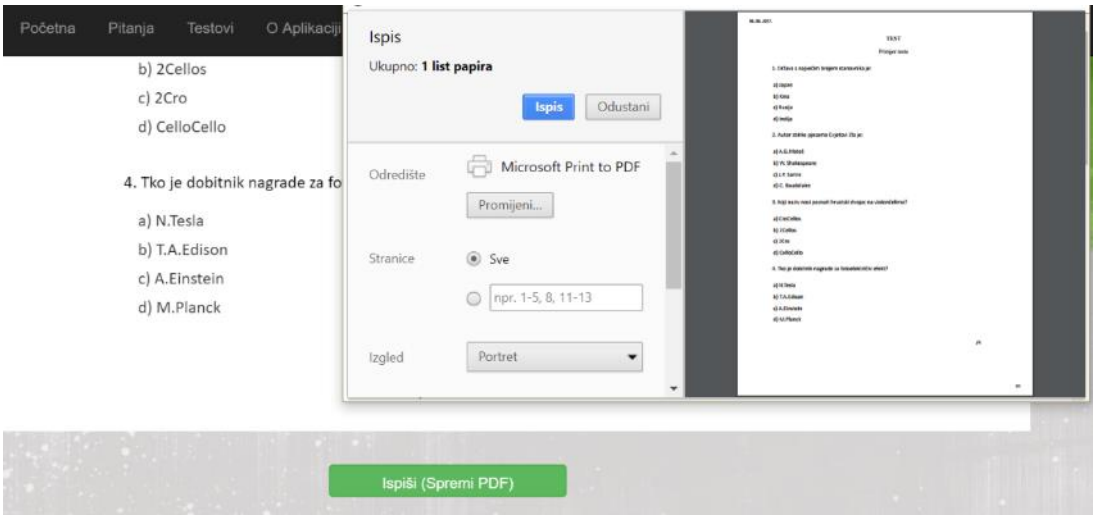
U slučaju da se ukupan broj pitanja ne podudara sa sumom pojedinačnih brojeva pitanja za svaku vrstu – također se ispisuje obavijest o greški. Nakon unosa potrebnih parametara i klikom na gumb *Generiraj test* uspješno se **generira test** te se otvaraju mogućnosti za novo generiranje s istim parametrima, generiranje s novim parametrima, prikaz testa i spremanje u bazu testova (Sl. 5.9.).



Sl. 5.9. *Izgled stranice nakon uspješno generiranog testa*

Generiranje s istim parametrima uzima iste parametre, ali su iz baze dohvaćena zaista drukčija pitanja od prethodnog generiranja. Odlaskom na *Zadaj nove parametre* stranica za generiranje se iznova pokreće te je moguće **unijeti parametre ispočetka**, a klikom na **prikaz testa** pokreće se nova kartica u kojoj se otvara generirani test (Sl. 5.10.). Na istoj se može vidjeti kako će test izgledati i koja su odabrana pitanja, a pri dnu korisnik ima opciju za **ispis** odnosno **spremanje u .pdf formatu** na računalo uz pomoć Microsoft Print to PDF opcije. (Sl. 5.11.).

Sl. 5.10. *Prikaz generiranog testa*

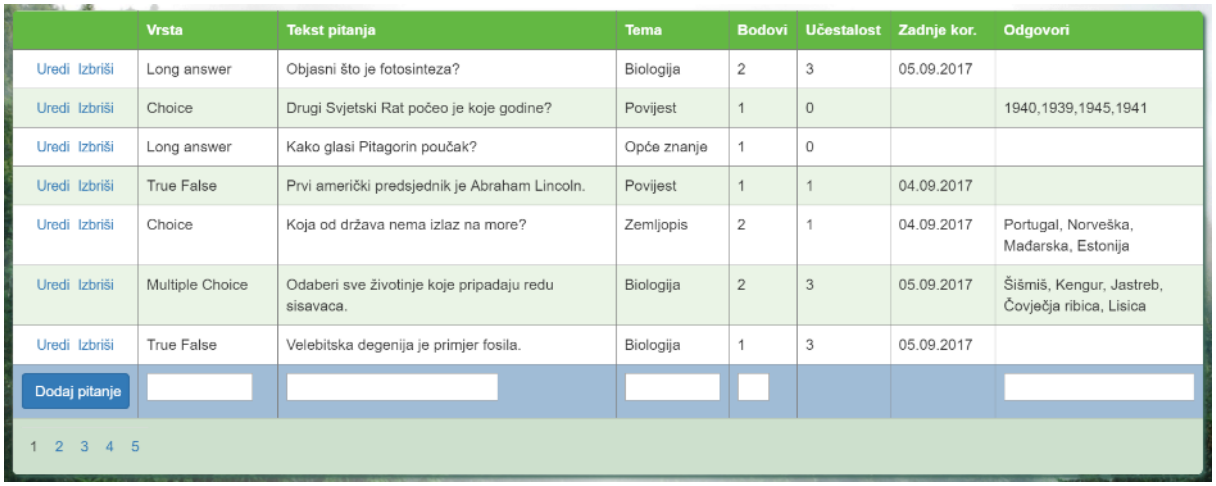


Sl. 5.11. *Pokretanje ispisa testa (ili spremanja u .pdf formatu)*

Klikom na *Spremi u bazu* test se **sprema u bazu** i ispisuje se poruka o uspješnom spremanju – isti je sada moguće vidjeti u bazi testova kao novi red u tablici.

5.2.2. Generiranje testa osobnim odabirom i baza pitanja

Pritiskom na gumb *Osobni odabir* unutar generatora testa ili odlaskom na opciju *Pitanja* u glavnoj izborničkoj traci, korisnik dolazi na stranicu za prikaz baze pitanja i osobni odabir pitanja za test. Ponajprije se testira opcija **rada s bazom pitanja**. Tablica (Sl. 5.12.) pokazuje sve bitne podatke za rad s pitanjima – vrsta, tekst, tema, bodove, koliko se često i kada zadnji put pojavilo pitanje te odgovore, ako ih ima.



	Vrsta	Tekst pitanja	Tema	Bodovi	Učestalost	Zadnje kor.	Odgovori
Uredi Izbriši	Long answer	Objasni što je fotosinteza?	Biologija	2	3	05.09.2017	
Uredi Izbriši	Choice	Drugi Svjetski Rat počeo je koje godine?	Povijest	1	0		1940,1939,1945,1941
Uredi Izbriši	Long answer	Kako glasi Pitagorin poučak?	Opće znanje	1	0		
Uredi Izbriši	True False	Prvi američki predsjednik je Abraham Lincoln.	Povijest	1	1	04.09.2017	
Uredi Izbriši	Choice	Koja od država nema izlaz na more?	Zemljopis	2	1	04.09.2017	Portugal, Norveška, Mađarska, Estonija
Uredi Izbriši	Multiple Choice	Odaberi sve životinje koje pripadaju redu sisavaca.	Biologija	2	3	05.09.2017	Šišmiš, Kengur, Jastreb, Čovječja ribica, Lisica
Uredi Izbriši	True False	Velebitska degenija je primjer fosila.	Biologija	1	3	05.09.2017	
Dodaj pitanje							

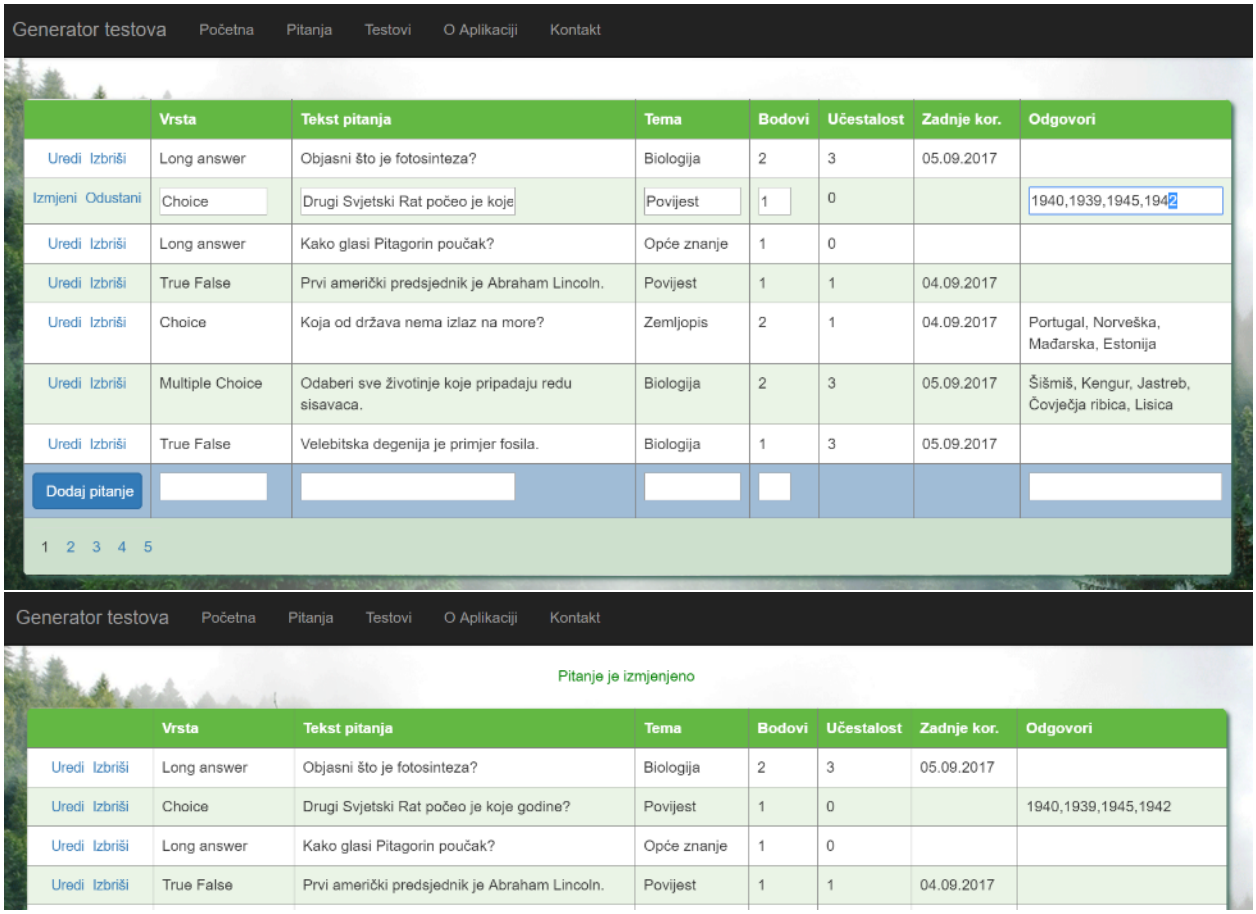
1 2 3 4 5

Sl. 5.12. *Baza pitanja*

U prvom stupcu omogućene su kontrole za **izmjenu** i **brisanje** pitanja, a pri dnu u podnožju tablice omogućen je **unos novog pitanja** s naznačenim mjestima unosa podataka i gumbom za

obavljanje unosa. Na samom kraju tablice, omogućeno je listanje pitanja pritiskom na željeni broj stranice.

Opcija uređivanja pitanja započinje stiskom na gumb *Uredi* pri čemu se javljaju kućice oko podataka u redu odnosno pitanju koje se želi izmijeniti, a u prvom stupcu javljaju se nove opcije za spremanje izmjene (*Izmjeni*) ili otkazivanje iste (*Odustani*). Za potrebe testiranja, izmjenjuje se drugo pitanje (red), točnije jedan njegov odgovor (Sl. 5.13.).

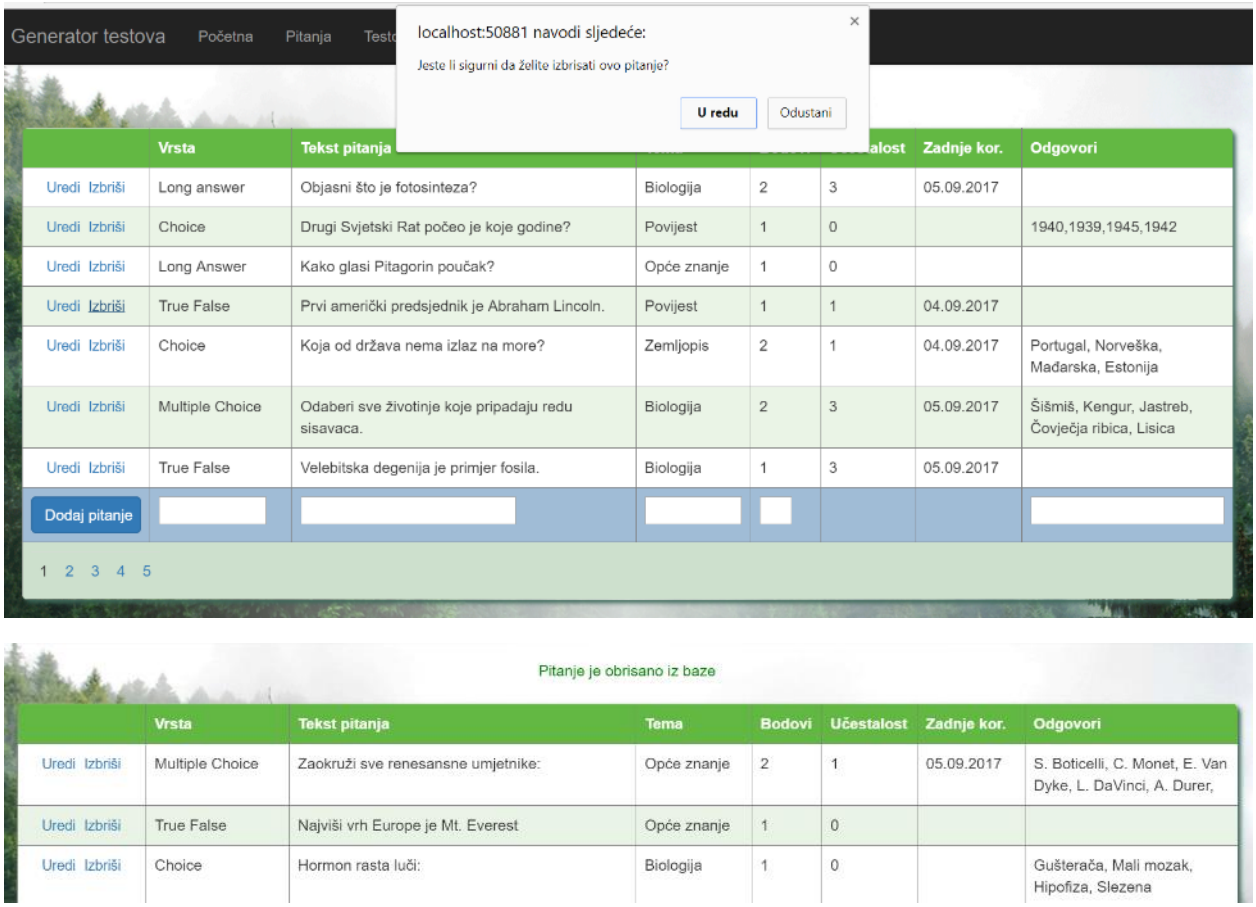


Sl. 5.13. *Uređivanje pitanja*

Klikom na *Izmjeni* pitanje je izmijenjeno i baza se vraća u prvobitno stanje, ali sa spremljenim promjenama (Sl. 5.13.), dok se opcijom *Odustani*, baza vraća u stanje prije uređivanja. Također je bitno napomenuti da će aplikacija spriječiti promjene koje ne odgovaraju formatima pitanja – npr. unos slova za broj bodova ili brisanje cijele liste odgovora te o istome ispisati i poruku.

Opcija brisanja pokreće se stiskom na *Izbriši* pri čemu se javlja poruka o potvrdi ili otkazivanju akcije (Sl. 5.14.) kako bi se izbjegla neželjena ili slučajna brisanja podataka od strane

korisnika. Potvrdom (*U redu*) pitanje nestaje iz baze, dok se odustajanjem vraća u dosadašnje stanje.

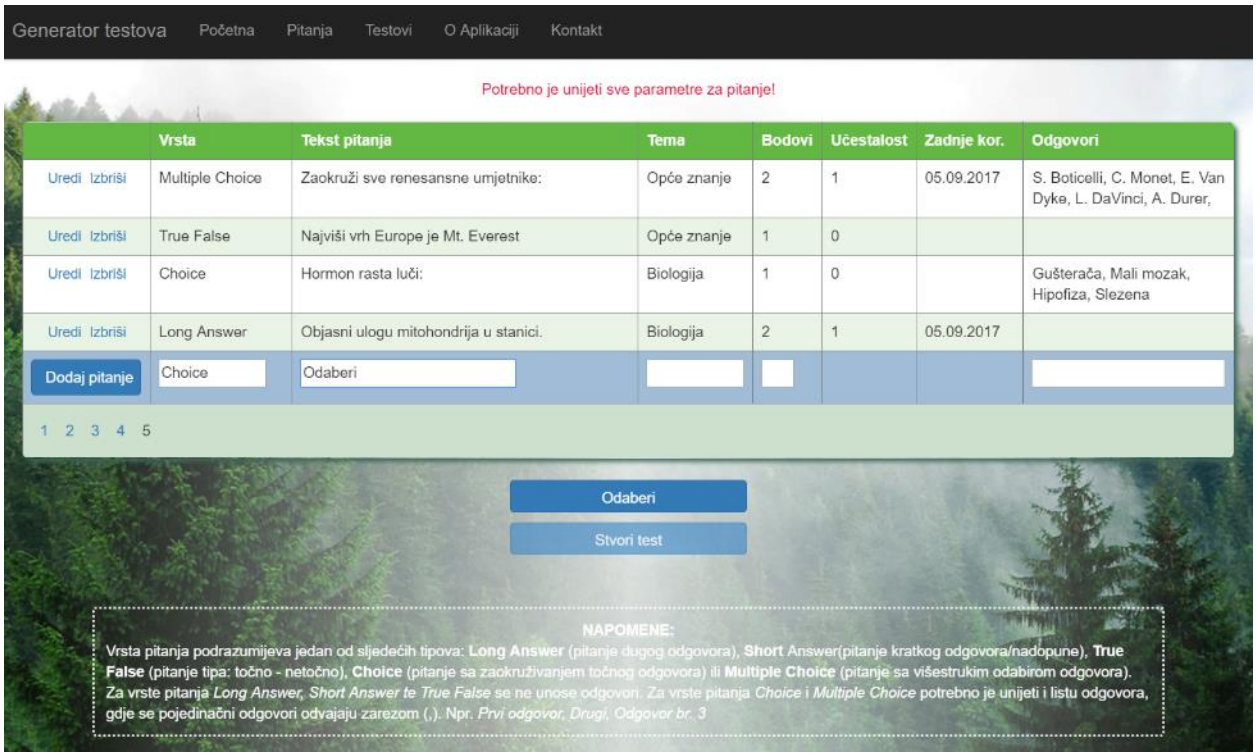


Sl. 5.14. *Brisanje pitanja*

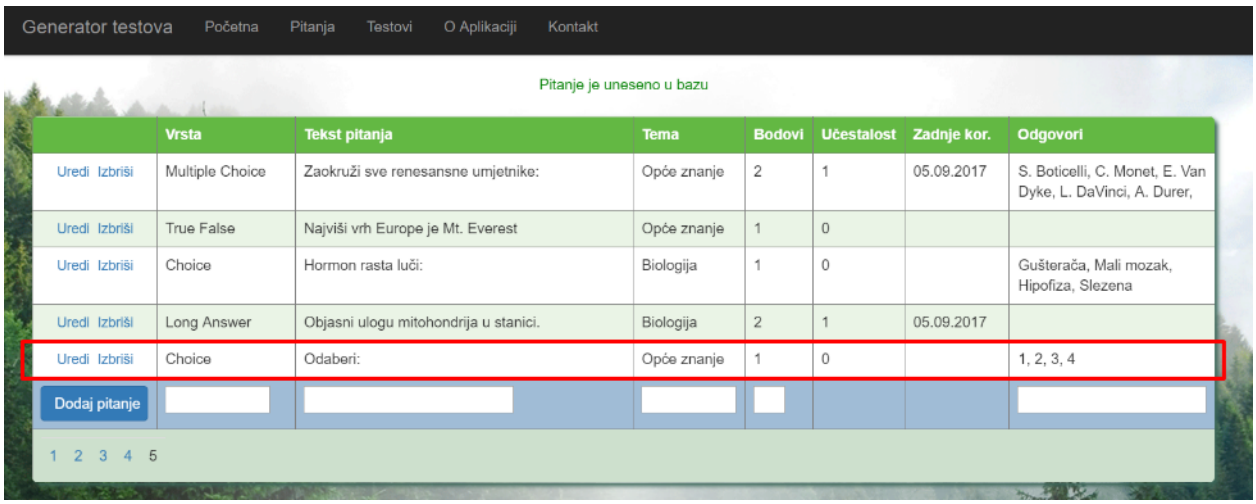
Unos novog pitanja obavlja se unosom podataka u kućice naznačenog plavog podnožja tablice te pritiskom na gumb *Dodaj pitanje*. U slučaju da neki od potrebnih parametara nije unesen, format vrste pitanja je u krivom obliku, za broj bodova su unesena slova ili odgovori nisu u pravilnom formatu – ispisuje se obavijest o greški i o čemu se radi, a na dnu se javljaju *Napomene* o pravilnom načinu unosa pitanja i funkcioniranja baze (Sl. 5.15.).

Nakon pravilnog unosa svih parametara, i klikom na gumb, pitanje je uspješno dodano u bazu, gdje se odmah može i vidjeti (Sl. 5.16.).

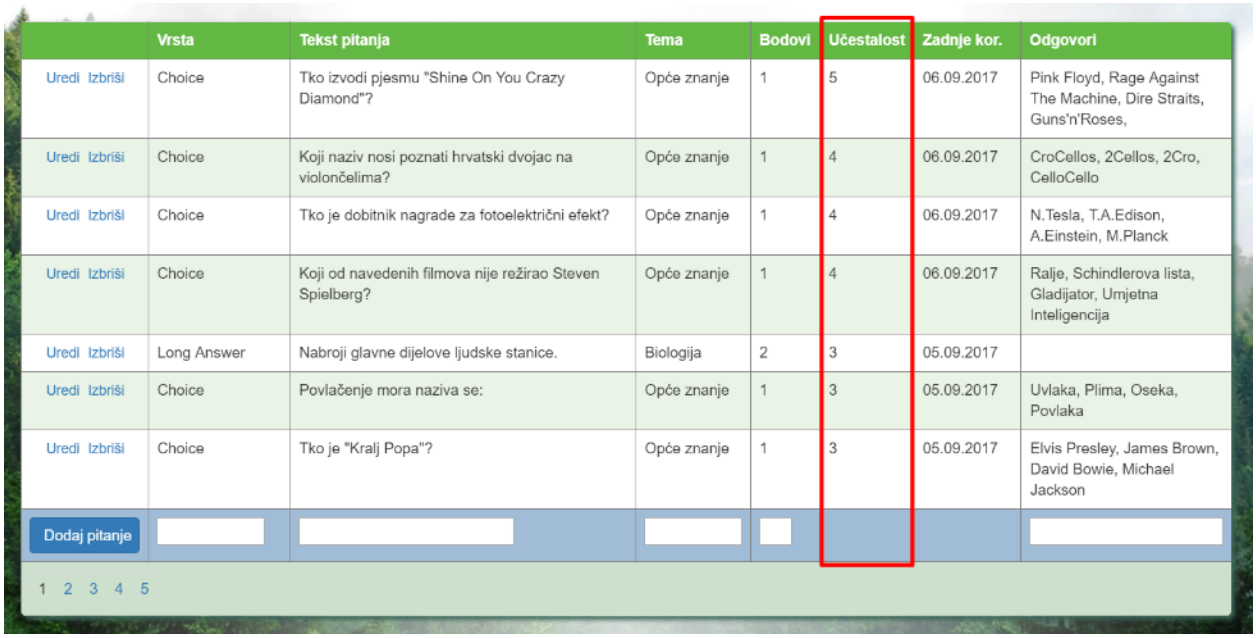
Uz to, za rad s bazom pitanja, omogućeno je i prebacivanje stranica tablice kao i sortiranje pitanja po nekom od parametara, klikom na naziv istoimenog stupca. Slika 5.17. prikazuje tablicu na prvoj stranici, sortiranu po učestalosti odnosno frekventnosti dosadašnjeg pojavljivanja u testovima.



Sl. 5.15. *Unos pitanja – nepravilan unos i Napomene*



Sl. 5.16. *Uspješno uneseno pitanje*



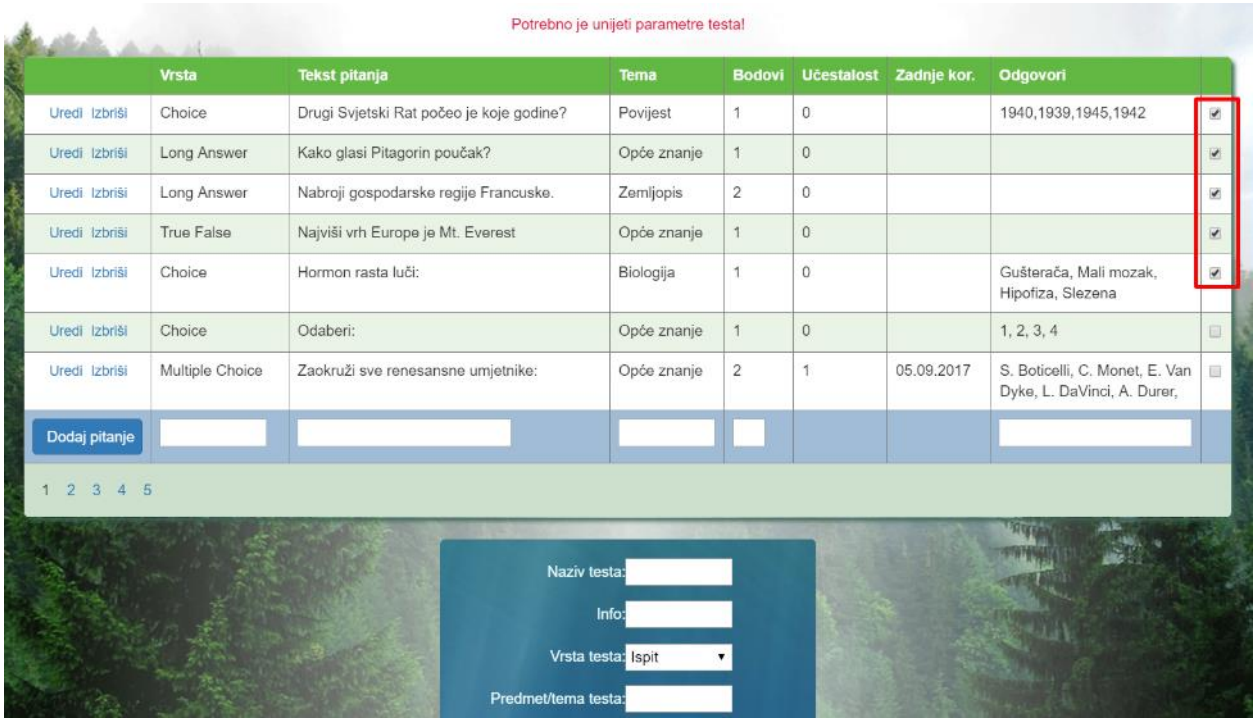
	Vrsta	Tekst pitanja	Tema	Bodovi	Učestalost	Zadnje kor.	Odgovori
Uredi Izbriši	Choice	Tko izvodi pjesmu "Shine On You Crazy Diamond"?	Opće znanje	1	5	06.09.2017	Pink Floyd, Rage Against The Machine, Dire Straits, Guns'n'Roses,
Uredi Izbriši	Choice	Koji naziv nosi poznati hrvatski dvojac na violončelima?	Opće znanje	1	4	06.09.2017	CroCellos, 2Cellos, 2Cro, CelloCello
Uredi Izbriši	Choice	Tko je dobitnik nagrade za fotoelektrični efekt?	Opće znanje	1	4	06.09.2017	N.Tesla, T.A.Edison, A.Einstein, M.Planck
Uredi Izbriši	Choice	Koji od navedenih filmova nije režirao Steven Spielberg?	Opće znanje	1	4	06.09.2017	Ralje, Schindlerova lista, Gladijator, Umjetna Inteligencija
Uredi Izbriši	Long Answer	Nabroji glavne dijelove ljudske stanice.	Biologija	2	3	05.09.2017	
Uredi Izbriši	Choice	Povlačenje mora naziva se:	Opće znanje	1	3	05.09.2017	Uvlaka, Plima, Oseka, Povlaka
Uredi Izbriši	Choice	Tko je "Kralj Popa"?	Opće znanje	1	3	05.09.2017	Elvis Presley, James Brown, David Bowie, Michael Jackson
Dodaj pitanje							

1 2 3 4 5

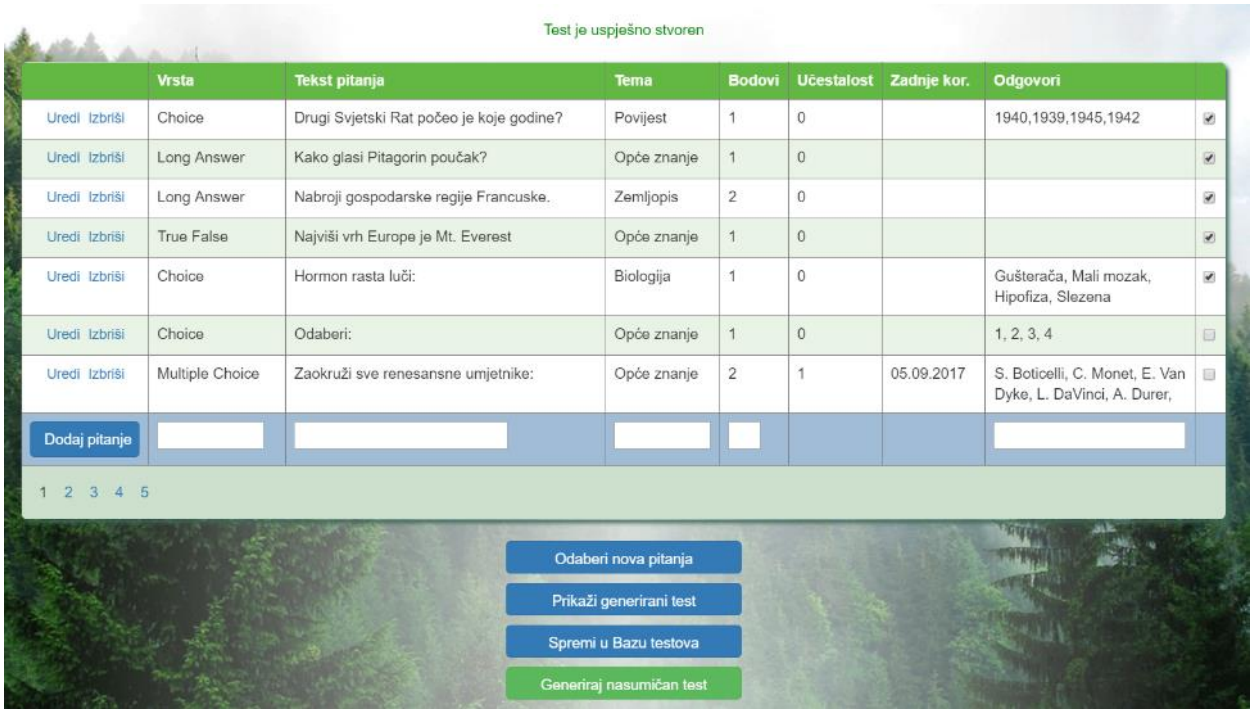
Sl. 5.17. *Pitanja poredana po učestalosti*

Sada se pristupa testiranju **generiranja testa s osobnim odabirom pitanja**. Moguće je odabrati bilo koja i koliko god pitanja je potrebno iz baze podataka za test. Za testiranje bit će potrebno generirati test koji će sadržavati 5 pitanja koja su se najrjeđe pojavljivala na dosadašnjim testovima. Prvo se započinje s odabirom pitanja stiskom na gumb *Odaberi* čime se u tablici stvara dodatni stupac za *CheckBoxove.* Pitanje se odabire klikom na odgovarajući *CheckBox* u istom redu. Zatim se stiskom na stupac *Učestalost* pitanja sortiraju uzlazno po učestalosti te se odabire potrebni broj pitanja. Prilikom klika na gumb *Odaberi* pojavila se i kućica za unos parametara testa (naziv, info, tema i sl.) koju je potrebno ispuniti prije samog stvaranja testa. U protivnom se ispisuje obavijest o istom. Ako su pak parametri ispunjeni, ali nijedno pitanje nije odabrano, porukom će se naznačiti da je potrebno odabrati i pitanja za test. Slika 5.18. prikazuje opisano.

Nakon unesenih parametara i odabranih pitanja, klikom na gumb *Stvori test*, aplikacija stvara test s odabranim pitanjima, prema zadanim parametrima te se o istom ispisuje poruka. Kućica za unos parametara nestaje, a pojavljuju se gumbi za opcije odabira novih pitanja, prikaza stvorenog testa, spremanja istog u bazu ili odlazak na generiranje nasumičnog testa ukoliko to korisnik želi (Sl. 5.19.).



Sl. 5.18. *Stvaranje testa – neispravni parametri*



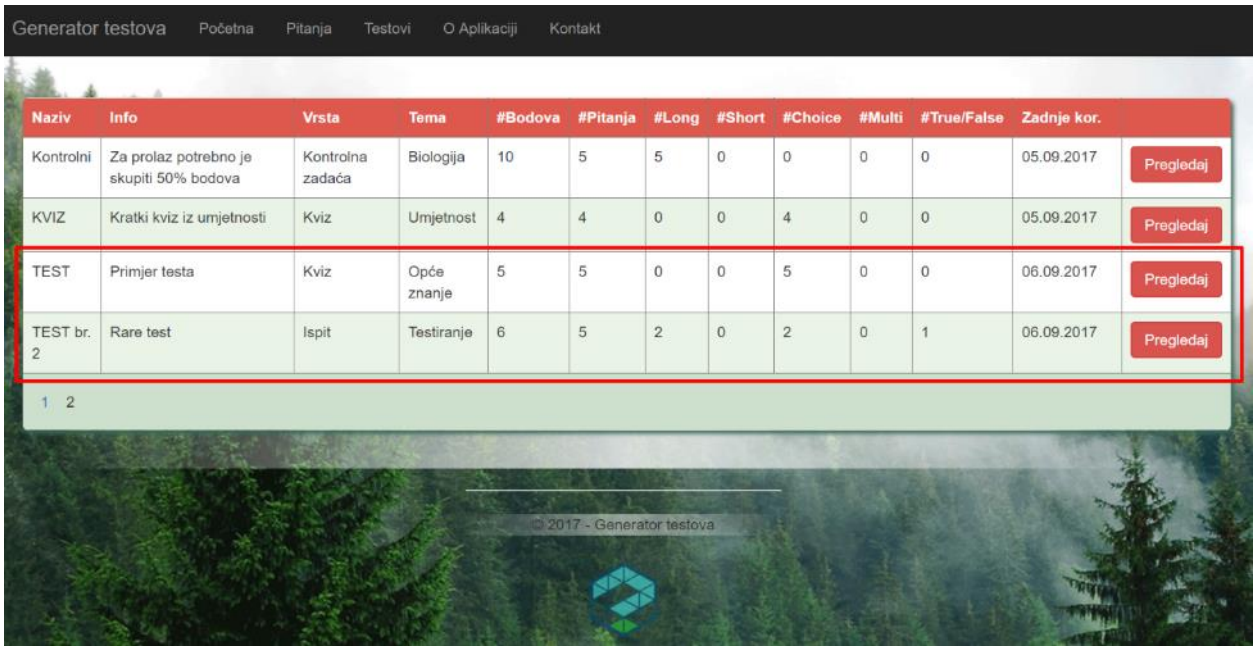
Sl. 5.19. *Stvaranje testa – uspješno stvoren test s osobnim odabirom pitanja*

Opcijom *Odaberi nova pitanja* iznova se započinje proces odabira pitanja, opcijom prikaži generirani test se identično kao kod nasumičnog generiranja test otvara u novoj kartici te se prikazuje njegov sadržaj, uz mogućnost ispisa/spremanja istog, dok se opcijom *Spremi u Bazu*

testova stvoreni test sprema u bazu korištenih testova, pri čemu se ispisuje poruka te se ažuriraju podaci o pitanjima koja su upotrebljena u testu, što se odmah može uočiti i kao promjena u tablici baze pitanja (učestalost im se povećala za 1, a kao datum zadnjeg korištena postavljen je trenutni datum). Sam test je sada dostupan u tablici baze testova te će isti biti tamo prikazan.

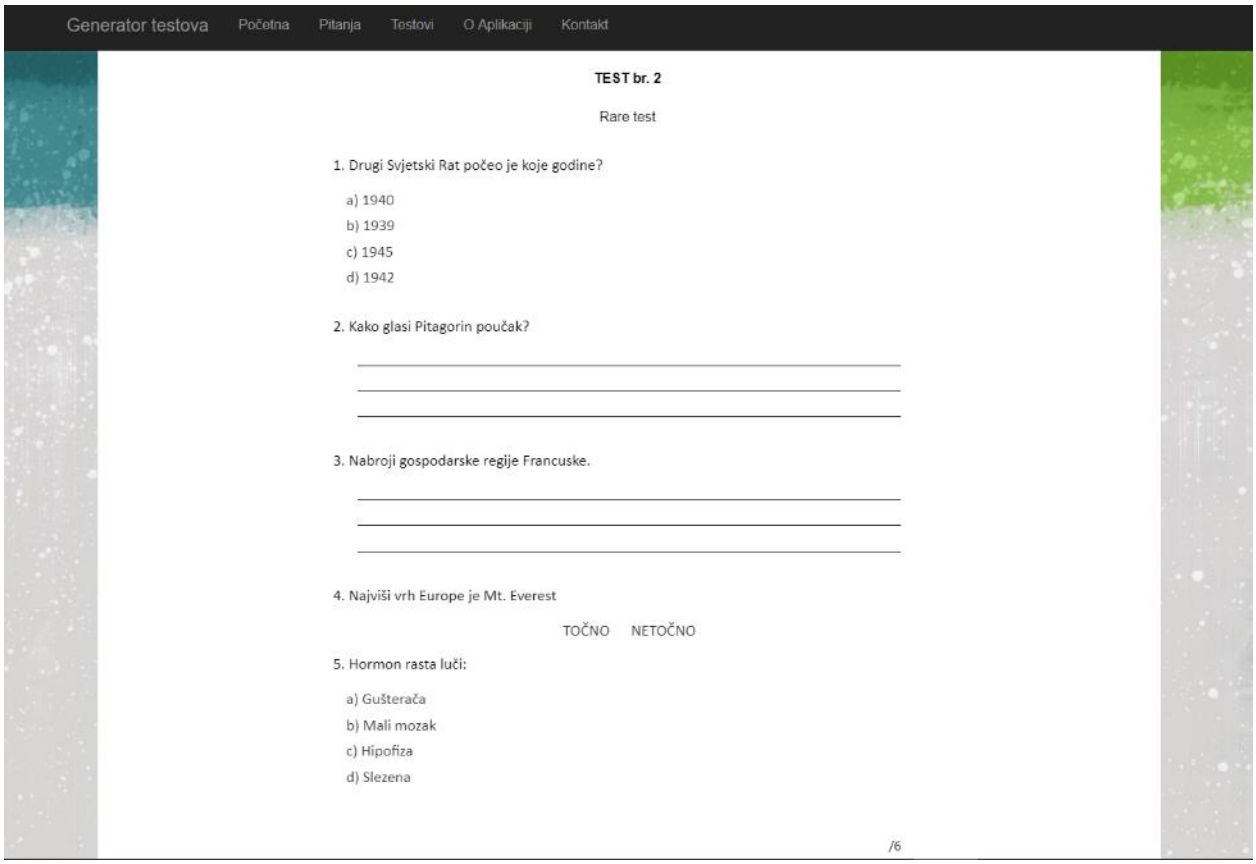
5.2.3. Baza testova

Odlaskom na opciju *Testovi* u glavnoj izbornoj traci, pristupa se bazi korištenih testova. Tablica baze testova sadrži sve potrebne informacije o testu (naziv, info, vrsta, broj bodova, broj pojedinačnih i ukupni broj pitanja, datum zadnje uporabe i sl.) uz omogućene opcije sortiranja. U zadnjem stupcu nalaze se gumbi za prikaz svakog od testova. U svrhu testiranja pronaći će se i prikazati upravo generirani testovi koji su bili spremljeni u bazu testova. Prelaskom na stranicu 2 prikazane tablice (Sl. 5.20.) kao zadnja dva unosa daju se uočiti upravo generirani testovi – nasumično generirani test (*TEST*) te test s osobnim odabirom pitanja (*TEST br. 2*).



Sl. 5.20. *Baza testova – dva testna primjerka*

Klikom na gumb *Pregledaj* pored testa *TEST* otvora se nova kartica s rekonstruiranim traženim testom kojeg je moguće ispisati ili spremiti u .pdf formatu. Klikom na gumb *Pregledaj* pored testa *TEST br. 2* također se otvara nova kartica u kojoj je prikazan traženi test (Sl. 5.21.).



Sl. 5.21. *Pregled korištenog testa (TEST br. 2)*

Dakle, aplikacija pokazuje zadovoljavajuće performanse: glede samih funkcionalnosti (nasumičnog generiranja, generiranja s osobnim odabirom pitanja, pristupa i radom s bazom pitanja, testova, prikaza, pregleda, spremanja i ispisa testa), brzine (u jako kratkom roku stvoren je i ispisan test), pouzdanosti izvođenja (više načina generiranja, mogućnost vlastitog odabira i sortiranja) te samog izgleda i dizajna jednostavnog *user-friendly* sučelja.

6. ZAKLJUČAK

Testovi raznih vrsta predstavljaju glavni oblik provjere znanja današnjice. Sa sve većim razvojem tržišta i napretka kako tehnologije, tako i čovječanstva, javlja se i potreba za agilnijim i efikasnijim testiranjima. *Quiz-builder* aplikacije (aplikacije za generiranje testova) omogućuju upravo to – skraćivanje vremena izrade testa, automatizaciju i jednostavnost cjelokupnog procesa, što za rezultat daje funkcionalne i kvalitetne testove.

Zadatak ovog završnog rada bio je realizirati i opisati upravo jednu takvu aplikaciju kojom bi se omogućilo generiranje testova, uz uvjet da bude efikasno, brzo te jednostavno za korištenje. Aplikacija se svojim svojstvima raznolikosti baze podataka, podrške raznih vrsta pitanja, načina pristupa i efikasnosti izrade izdvaja od sličnih aplikacija što joj u određenoj mjeri otvara put za daljnji razvoj i komercijalne primjene.

Aplikacija je realizirana uz pomoć Microsoft Visual Studia, koristeći pri tome bazu kreiranu uz pomoć Microsoft SQL Servera. Uz poznavanje C#, .NET tehnologije (prvenstveno ASP.NET web aplikacija), HTML-a i sličnih potrebnih tehnologija i programskih jezika, aplikacija je uspješno realizirana te ispunjava početne zahtjeve. Međutim, kao i uvijek, postoji mnogo mjesta za poboljšanje iste – prvenstveno grafičkog izgleda i dizajna same aplikacije, dodavanje novih funkcionalnosti u aplikaciju (npr. dodatni parametri generiranja, kalkulator optimalnih parametara itd.), daljnjog poboljšanja performansi i sl.

LITERATURA

[1] Microsoft Visual Studio: Wikipedia, 2017., dostupno na: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio [25. lipnja 2017.]

[2] SQL Server Management Studio: Wikipedia, 2017., dostupno na: https://en.wikipedia.org/wiki/SQL_Server_Management_Studio [26. lipnja 2017.]

[3] C Sharp (programming language): Wikipedia, 2017., dostupno na: https://en.wikipedia.org/wiki/C_Sharp_(programming_language) [26. lipnja 2017.]

[4] Fakultet elektrotehnike računarstva i informacijskih tehnologija Osijek, Programiranje 2: LV12 - Osnove objektno orijentiranog programiranja i uvod u C#, str.3, 2017.

[5] .NET Framework: *NET Framework component stack*, Wikipedia, 2017., dostupno na: https://en.wikipedia.org/wiki/.NET_Framework [04. rujna 2017.]

SAŽETAK

Cilj ovog završnog rada bio je realizirati aplikaciju za generiranje testova koja će na korisnikov zahtjev, po unesenim parametrima, kreirati test sa slučajno ili osobno odabranim pitanjima. Korišten je programski jezik C# te ASP.NET, Entity Framework, CSS te JavaScript tehnologije uz pomoć programskih paketa Microsoft Visual Studio 2017 te Microsoft SQL Server 2016. Ponajprije je stvorena potrebna baza podataka potrebna za realizaciju aplikacije. Nakon povezivanja s bazom podataka, pristupilo se izradi pojedinih dijelova aplikacije (stranica). Navedeno je realizirano pisanjem označnog koda, C# koda u pozadini stranice te primjenom potrebnih skripti (JavaScript) i CSS koda za dizajn prikaza. Kao rezultat dobivena je web aplikacija kojom je moguće uz bazu pitanja učinkovito i praktično generirati test brzim i jednostavnim slučajnim odabirom pitanja ili osobnim odabirom po željenim preferencijama, uz skraćeno ukupno vrijeme sastavljanja i ispisa testa.

Ključne riječi: Web aplikacija, C#, ASP.NET, Microsoft Visual Studio, generator testova.

C# APPLICATION FOR GENERATING TESTS

The goal of this project was to create an application for generating tests that would create randomly generated or personally formed tests, on user's demand, according to the input parameters. In order to develop such application, C# programming language, ASP.NET, Entity Framework, CSS and JavaScript technologies have been used, with the help of Microsoft Visual Studio 2017 and Microsoft SQL Server 2016 developing softwares. The first step was to create the necessary database for the realization of the application. After the connection with the database was established, the following step was to create individual parts of the application (web pages). The previously mentioned has been realised by writing the mark-up code, the C# code behind and the required scripts (JavaScript) and CSS code for the design of the web pages. The result is a web application that uses database of questions and enables efficient and practical generating of tests whether by a rapid and simple random selection or by a personal selection of questions in alignment with the wanted preferences, which results in a shorter test creation and printing time.

Keywords: Web application, C#, ASP.NET, Microsoft Visual Studio, test generator.

ŽIVOTOPIS

Autor ovog završnog rada, Vedran Ivić, student je treće godine preddiplomskog studija Elektrotehnike na Fakultetu Elektrotehnike, Računarstva i Informacijskih Tehnologija i trenutno živi u Odžaku (BiH). Rođen je 09.10.1995. u Münchenu (Njemačka), a 2002. započinje svoje osnovnoškolsko obrazovanje u Odžaku koje završava kao učenik generacije s istaknutim uspjesima na natjecanjima iz matematike te fizike. Godine 2010. upisuje opću gimnaziju u Srednjoj školi Pere Zečevića u Odžaku, gdje uz sudjelovanje i ostvarene uspjehe na natjecanjima te izvanškolskim aktivnostima ponovno postaje učenik generacije. U Osijeku, 2014. godine upisuje preddiplomski studij elektrotehnike na tadašnjem Elektrotehničkom Fakultetu, na kojem se odlučuje za smjer Komunikacije i Informatika. Kao student FERIT-a, držao je demonstrativnu nastavu na više kolegija te sudjelovao na više međunarodnih natjecanja studenata elektrotehnike (Elektrijada), a od 2016. član je međunarodne organizacije mladih AIESEC, gdje postaje i potpredsjednik Lokalnog Odbora AIESEC Osijek. Kroz fakultetsko obrazovanje iz godine za godinom ostvaruje odličan uspjeh, a 2017. dobiva i dekanovu nagradu kao najbolji student na svome smjeru.

Vedran Ivić

PRILOZI

CD:

- Elektronička verzija rada (.doc i .pdf format)
- Visual Studio projekt web aplikacije