# **UČILICA - PROGRAMERSKA DOKUMENTACIJA**

Izradili : Ivona Čižić, Luka Jandrijević, Helena Marciuš, Dora Raštegorac

Link na github repozitorij: <https://github.com/jandra1/ucilica>

Cijeli projekt je izrađen koristeći C# programski jezik, uz pomoć Windows formi i SQL baza podataka. Za pokretanje aplikacije potrebno je instalirati Visual Studio, download link se može pronaći na sljedećem linku: https://visualstudio.microsoft.com/ (potrebno je skinuti .NET framework od ponuđenih paketa) i softver je besplatan. Da bi vam radilo spajanje na bazu podataka, možda ćete morati skinuti i instalirati program sa sljedeće stranice: https://www.microsoft.com/en-us/download/confirmation.aspx?id=13255 . Ako i dalje imate problema sa spajanjem na bazu podataka pri paljenju aplikacije, unutar “properties” od projekta, maknite kvačicu sa “prefer 32 bit”.

U datoteci “pitanja i slike” nalaze se pitanja i pripadajuće slike koje idu uz pitanja koja se nalaze u Učilici. Unutra su pitanja i odgovori koji će se prikazivati u aplikaciji. Pitanja su za 5., 6., 7. i 8. razred osnovne škole za predmete: hrvatski jezik, geografiju, povijest, matematiku i engleski jezik.

U opis-projekta.docx se može pronaći malo detaljniji opis kako bi aplikacija trebala raditi i čemu služi. U datoteci Ucilica se nalazi Ucilica.sln kojeg treba pokrenuti za početak rada aplikacije.

## **Ucilica/Ucilica**

U datoteci Ucilica/Ucilica nalaze se svi potrebni file-ovi za rad aplikacije. Properties je datoteka koja se automatski generirala. U Resources možemo pronaći sve potrebne slike koje se koriste te audio zapise koji se puštaju tijekom korištenja aplikacije. Sve nove slike, audio zapise i slično, potrebno je tu staviti.

Svi file-ovi koji imaju nastavak Designer.cs ili .resx su fileovi koji se generiraju kada se na formu dodaju gumbi ili ostale komponente te se generiraju od strane Visual Studia.

Početna forma je Form1.cs u kojoj imamo string name u kojem čuvamo podatke za username pri logiranju u aplikaciju. Također, u njoj se spajamo na bazu podataka koja čuva username i password. Prilikom klika na gumb login ili tipku Enter poziva se funkcija Login\_Click koja obrađuje korisnikov unos te zahtijeva da upiše username i lozinku koja može sadržavati samo brojeve. Nakon toga, funkcija se pokušava spojiti na bazu podataka koja sadrži login podatke o pojedinom korisniku. Ako postoji točno jedna osoba u bazi podataka koja zadovoljava uvjete zatvaramo vezu s bazom i prelazimo na glavnu stranicu. Ako se prilikom ulogiravanja prijavio admin, s usernameom admin i lozinkom 1111, otvara se forma drugačija od forme za običnog korisnika.

Ako korisnik klikne na gumb za zatvaranje aplikacije, tada se forma gasi, a ako klikne na registraciju, otvara se posebna forma za registraciju.

Admin.cs je forma kojoj smije pristupiti samo admin, tj. osoba koja održava igru te preko gumba na njoj pristupa formi za dodavanje novih pitanja.

Dodaj.cs je forma u kojoj admin dodaje nova pitanja u bazu. U klasi se nalazi lista predmeti koja sadrži sve predmete koji su dostupni u aplikaciji: geografija, matematika, hrvatski, povijest i engleski te, kada se klikne na gumb, spajamo se na bazu podataka i dodajemo pitanje, odgovore i razred koji su upisani od strane admina u odgovarajuće TextBox-ove.

U formi Profil korisnik vidi svoje podatke, rezultate te može promijeniti ime, lozinku i razred.

U klasi Kraj, definiramo formu koja prikazuje vrijeme i finalni broj bodova.

Klasa Bodovi.cs nam služi za prikazivanje bodova svih korisnika koji su postigli rezultat igrajući učilicu i to poredano krenuvši od najboljeg, tj. korisnika koji ima najviše bodova. Ako se u ComboBox-u za predmet ili razred ne nalazi prazan tekst, tada pozivamo funkciju changeTable() koja se brine o ispravnom prikazu svih rezultata na formi.

Pitanje.cs je dio forme koji sadrži varijable odgovorne za prikazivanje pitanja iz baze podataka. U konstruktoru inicijaliziramo razred, predmet te zvukove yay i wrong. Također pozivamo funkciju sljedećePitanje() koja prikazuje iduće pitanje iz baze podataka. Postoje 4 moguća odgovora, pa će tako postojati i 4 funkcije koje se pozivaju na klik te svaka od njih provjerava je li kliknut odgovor točan, ako je, gumb se zazeleni, svira određena melodija i broj bodova se povećava. U protivnom se gumb zacrveni i svira zvuk za krivi odgovor te se prikazuje točan odgovor tako što se pripadni gumb zazeleni.

Funkcija sljedećePitanje() je odgovorna za odabir sljedećeg pitanja iz baze podataka. Prikazujemo stanje bodova na svakom idućem pitanju te pomoću randomizacije odabiremo iduće pitanje iz baze. Ako pitanje sadrži sliku, i nju prikažemo u formi. Također, u formi se prati na kojem smo pitanju.

U funkcijama initializeTime() i t\_Tick() brinemo o otkucaju sata te tako mjerimo vrijeme u sekundama i zapisujemo ga. onemoguciGumbe()/omoguciGumbe() postavlja gumbe za odabir odgovora na false/true dok u funkciji hintButton\_Click() prikrivamo jedno pogrešno pitanje kada korisnik pritisne tipku za hint. U toj funkciji prvo gledamo koji odgovor je krivi, te taj odgovor postavljamo na crnu boju i uz pomoć atributa tog gumba onemogućimo klik na njega.

Forma Razred.cs je prva forma koja se prikazuje nakon što se obični korisnik ulogira. U njoj se spajamo se na bazu podataka i čuvamo listu svih predmeta. Njih zatim prikazujemo u ComboBox-u. Klikom na gumb "Kreni s učenjem!", prikazujemo formu pitanje i to tako da joj prosljeđujemo odabran predmet kako bi kasnije znali iz kojeg predmeta izvlačimo pitanja. Također, korisnik ima mogućnost odjaviti se (log out), vidjeti rang listu svih igrača i otići na svoj profil. Na vrhu postoji gumb koji služi za mute/unmute zvukova koji nastaju tijekom igranja igre.

U Register.cs klasi brinemo se o registraciji novih korisnika. Na početku se spajamo na bazu te postavljamo veličinu forme u konstruktoru. Klikom na gumb za registraciju prvo provjeravamo unos od strane korisnika. Polje za username i lozinku ne smije biti prazno te lozinka mora sadržavati minimalno 4 broja, bez slova. Zatim se pokušavamo spojiti na bazu i u bazi pretražujemo je li zadani username koji je korisnik upisao već postoji. Ako postoji, obavještavamo korisnika i ne radimo ništa. Taj postupak ponavljamo i za lozinku. Ako je sve ostalo u redu, dodajemo novog korisnika u bazu podataka.

Klasa database.cs služi za izvlačenje pitanja iz baze. Ona ima funkciju getQuestionsByYearAndSubject() koja pomoću baze podataka dohvaća pitanja iz određenog razreda i predmeta. Izvuče sve moguće odgovore, slike i pitanja. Funkcija addQuestion() služi za dodavanje pitanja u bazu podataka pomoću SQL naredbi. Također, postoji funkcija getScoresBySubjectAndYear() u kojoj dohvaćamo, iz baze podataka, top 10 korisnika s najvećim brojem bodova za određeni razred i predmet. U ovoj klasi brinemo se i za upis novog broja bodova u bazu pa za to postoji funkcija insertNewScore() u kojoj u tablicu "bodovi" dodajemo novog korisnika s određenim brojem bodova i vremenom u kojem je završio kviz. Funkcija getResultsByName() dohvaća sve rezultate određenog korisnika. Funkcije changeUserName(), changePass() i changeYear() služe redom za promjenu korisničkog imena, lozinke i razreda određenog korisnika.

Klasa pitanjeKlasa.cs služi za spremanje i čuvanje pitanja i odgovora za odabrane predmete i razrede te za čuvanje informacija o slici i razredu kojem je pitanje namijenjeno. Ovu klasu koristimo na više mjesta.

Što treba popraviti ili dodati?

- više pitanja za ponuđene razrede i predmete

- više mogućnosti za admina

- dodati preostale razrede i predmete osnovne škole te proširiti na srednju školu

- više pitanja sa slikom ili videom

- kada je kviz pokrenut, moguće se logout-ati i nastaviti igrat kviz. Možda bi se moglo staviti da, kada se stisne logout za vrijeme kviza, svi prozori se zatvaraju.

- omogućiti kviz s pitanjima iz više predmeta (korisnik sam bira predmete)

- omogućti rješavanje kviza bez login-a (tada se rezultat ne sprema u bazu)

- dodati da svira neka pjesma tijekom igre.

- rang ljestvica s poretkom korisnika u kojoj se i vrijeme završetka uzima u obzir