**OʻZBEKISTON RESPUBLIKASI**

**OLIY TAʼLIM, FAN VA INNOVATSIYALAR VAZIRLIGI**

**MIRZO ULUG‘BEK NOMIDAGI**

**OʻZBEKISTON MILLIY UNIVERSITETININIG JIZZAX FILIALI**

****

**AMALIY MATEMATIKA FAKULTETI**

**«KOMPYUTER ILMLARI VA DASTURLASHTIRISH» kafedrasi**

**“ALGORITMIK TILLAR VA DASTURLASH” FANIDAN**

**KURS ISHI**

**Mavzu:** ***Savollar bankidan fan bo‘yicha imtihon biletlarini yaratish imkonini beruvchi avtomatlashtirilgan axborot tizimini ishlab chiqish***

**Bajardi:** 60610100 – Kompyuter ilmlari va dasturlash texnologiyalari ta’lim yoʻnalishi 2 -kurs 472-22 - guruh talabasi

**Axmedova Feruza G‘ayrat qizi**

**Kurs ishi rahbari: Jomurodov D.M.**

**Jizzax – 2024**

| **№** | **Kurs ishi boʻyicha ish turlari** | **Ball** |
| --- | --- | --- |
| **1** | **Kurs ishini hajmi va talab boʻyicha rasmiylashtirilganligi:**  - talabga javob beradi (7-8);  - talabga qisman javob beradi (6-7);  - talabdan chetga chiqish xolatlari mavjud (5-6);  - talabga javob bermaydi (0-4); |  |
| **2** | **Mavzuning dolzarbligining asoslanganligi:**  - yetarli darajada asoslanganligi (7-8);  - yetarli darajada asoslanmagan (6-7);  - noaniq (5-6);  -asoslanmagan (0-4); |  |
| **3** | **Maqsad vazifalarining aniq ifodalanganligi:**  - aniq (7-8);  - toʻliq aniqlanmagan (6-7);  - aniq emas (5-6);  - ifodalanmagan (0-4) |  |
| **4** | **Kurs ishining himoya qismida ijtimoiy-iqtisodiy (ta’lim) sohasida qoʻllashga tavsiya berilganligi:**  - bevosita qoʻllashga tavsiya bor (7-8);  - qisman qoʻllashga tavsiya qilingan (6-7);  - tavsiya ishonchli emas (5-6);  - tavsiya yoʻq (0-4); |  |
| **5** | **Olingan natijalarni yangiligi va ishonchlilik darajasi:**  - natija yangi (7-8);  - ilgari olingan (6-7);  - toʻla ishonchli emas (5-6);  - eskirgan, ishonchli emas (0-4); |  |
| **6** | **Talabaning mavzu boʻyicha olingan natijalarini tanqidiy baholanganligi darajasi:**  - aniq (7-8);  - toʻliq aniq emas (6-7);  - juda sayoz (5);  - tanqidiy baholanmagan (0-4); |  |
| **7** | **Adabiyotlardan foydalanganlik darajasi:**  - qonunlar, qarorlar, ilmiy-amaliy jurnallar, monografiya, yetakchi olimlar asarlaridan toʻla foydalanilgan (7-8);  - darslik, ma’ruza matnlari, oʻquv-qoʻllanma va ma’lumotlardangina foydalanganligi (6-7);  - yetarli foydalanilmagan (5);  - foydalanilmagan (0-4); |  |
| **8** | **Talabaning ma’ruzasiga baho:**  - a’lo (14-15);  - yaxshi (11-13);  - qoniqarli (9-10);  -qoniqarsiz (0-8) ; |  |
| **9** | **Talabaga berilgan savollarga javobi:**  - toʻliq (14-15);  - yaxshi (11-13);  - qoniqarli (9-10);  -qoniqarsiz (0-8); |  |
| **10** | **Adabiyotlar tanqidiy taxlili darajasi:**  - toʻla darajada (7) ;  - qisman darajada (5-6);  - noaniq (4-5);  - taxlil yoʻq (0-3); |  |
| **11** | **Internet manbalaridan foydalanganlik idarajasi:**  - yetarli, natija yangi (7);  - qisman, ilgari olingan (5-6);  - toʻla ishonchli emas (4-5);  - foydalanilmagan (0-3); |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Toʻplangan ball (bahosi):** | | |  |
|  | | |  |
| **Komissiya raisi:** | | | |
|  |  |  | *KAYUMOV OYBEK ACHILOVICH* |
|  | *(imzo)* |  |  |
|  | **Komissiya a’zolari:** | | |
|  |  |  | *BARATOV JASUR RUSTAM O‘G‘LI* |
|  | *(imzo)* |  |  |
|  |  |  | *ULASHEV ASRORJON NASRIDINOVICH* |
|  | *(imzo)* |  |  |
|  |  |  | *IRGASHEVA UMIDA ABDIMITAL QIZI* |
|  | *(imzo)* |  |  |
|  |  |  | *MAHKAMOV SHOHRUH SARVAR O‘G‘LI* |
|  | *(imzo)* |  |  |
| **Kurs ishi rahbari:** | | | |
|  |  |  | *JOMURODOV DUSTMUROD MAMASOLI O‘G‘LI* |
|  | *(imzo)* |  |  |
|  |  |  |  |

MUNDARIJA

[KIRISH 5](#_Toc166658650)

[I-BOB. AVTOMATLASHTIRILGAN AXBOROT TIZIMINI ISHLAB CHIQISHNING NAZARIY ASOSLARI 8](#_Toc166658651)

[1.1. Tizimni yaratishda qo`llanilgan texnik va dasturiy vositalar haqida malumot va ularning imkoniyatlari 8](#_Toc166658652)

[1.2. Tizim uchun foydalanilgan berilganlar bazasi haqida 21](#_Toc166658653)

[II BOB. AVTOMATLASHTIRILGAN AXBOROT TIZIMINI YARATISH IMKONINI BERUVCHI DASTURIY TA’MINOT YARATISH 24](#_Toc166658654)

[2.1. Loyihaning yaratilish qismi bilan tanishish 24](#_Toc166658655)

[2.2. Yaratilgan tizimining amaliy ahamiyati va qo’llanish sohalari 29](#_Toc166658656)

[XULOSA 32](#_Toc166658657)

[FOYDALANILGAN ADABIYOTLAR 33](#_Toc166658658)

[ILOVALAR 34](#_Toc166658659)

# KIRISH

Jamiyatni axborotlashtirish hozirgi zamonning davr talabiga aylanib bormoqda. Inson faoliyat olib borayotgan barcha sohalar axborotni qabul qilish, qayta ishlash va o‘zlashtirish jarayonlari bilan uzviy bog‘liq.Bugungi kunda tashkilot qanday bo‘lishidan qat’iy nazar ular moliyaviy bo‘ladimi, savdo tashkilotlari yoki sanoat bo‘ladimi barchasining faoliyatini berilganlar bazasisiz tasavvur qilib bo‘lmaydi.

**Mavzuning dolzarbligi** Bu tizimning muhim qismi, savollar va fanlar orasidagi ta'alluqni aniqlab berish va savollarni imtihonda maqsadga muvofiq tuzishdir. Dolzarbligi tizimining foydalanuvchilarga mos ravishda savollarni berish, maslahat bermoq va ularni imtihon natijalarini chiqarishga yordam berish kerak va o'quv jarayonini samarali va tushunarli qilishga yordam beradigan avtomatlashtirilgan axborot tizmini (AIT) ishlab chiqishning ahamiyatini ifodalaydi. Bu tizim, o'quvchilarning bilimini oshirish, o'quv jarayonini tashkil etish, ma'lumotlarni to'plash va saqlash, natijalarni hisoblash va analiz qilish, shuningdek o'quvchilarning o'zgartirishlarini kuzatish va adaptiv o'quvni amalga oshirishga yordam beradi. AITning ishlab chiqilishi, o'quvchilarning o'zlarining o'quv jarayonini boshqarishga imkon beradi, shuningdek o'qituvchilarning va o'quvchilarning vaqtini samarali ishlatishga yordam beradi.Kurs ishini aniqlash, loyihaning barcha qismlarini va ularning o'zaro bog'liqligini tahlil qilishni o'z ichiga oladi.Shuningdek, loyihaning samaradorligi va ishonchli ishlashi uchun kerak bo'lgan o'zgarishlar va o'zgartirishlar aniqlanadi.Bu jarayon, loyihaning barcha qismlarini va ularning o'zaro bog'liqligini tahlil qilish, loyihaning o'zgarishlarini kuzatish va boshqarish, loyihaning samaradorligini optimallashtirish va ishonchli ishlashini ta'minlash uchun kerak bo'lgan sinovlar va testlarni aniqlashni o'z ichiga oladi.

**Maqsadlar va vazifalari:**

Imtihon Savollarini Avtomatlashtirish: Bu texnologiya imtihon savollarini avtomatlashtiradi, bu esa tizimni o'rganish va o'qituvchilar uchun vaqt va energiyani tejashga yordam beradi.Tizim, foydalanuvchilar uchun mos ravishda fanlarni va savollar bilan bog'liq ta'limotlarni aniqlash uchun ma'lumotlarni to'plab boradi.

O'rganuvchilar Uchun Maxsus Imtihon Tuzish: Tizim foydalanuvchilarga shaxsiy ta'lim jadvalini tuzish imkonini beradi va ularning o'zaro munosabatlarni taqdim etadi.

Natijalarni Tahlil Qilish va Ko'rsatish: Tizim imtihon natijalarini tahlil qiladi va o'rganuvchilarga shaxsiy ta'lim rejalarini takomillashtirish uchun tavsiyalar beradi.

Ushbu avtomatlashtirilgan axborot tizimi, o'rganuvchilarni tayyorlash, o'qituvchilarga qulayliklar taqdim etish va o'rganish jarayonini ijobiy yo'lda muvaffaqiyatga olib borish maqsadida yaratilgan. Bu tizimning dolzarbligi, imtihonlar va o'qitish jarayonlarida samarali yordam berishda namoyon bo'ladi.

Savollar va Fanlar Menejmenti: Tizim, savollar va fanlar bazasini menejerlik qiladi. Bu bazada savollar va ularga oid ma'lumotlar ( fan bo`yicha testlar to`plami, vaqt chegaralari, kategoriyalar, va boshqalar) saqlanadi.

Imtihon Natijalarining Tahlili: Imtihon natijalarini tahlil qilish va tizim orqali foydalanuvchilarga ma'lumotlar berish. Bu, o'rganuvchilar va o'qituvchilar uchun o'quv jarayonini baholashda yordam beradi.Tizimda bazada saqlangan savollar bo`yicha tashkil qilingan testni o`quvchilar ishlashi orqai ularning natijalari hamda qaysi vaqtda qanaqa fandan test topshirganligini o`qituvchi nazorat ostida saqlab turadi.

Ma'lumotlar Xavfsizligi: Tizim ma'lumotlarni xavfsiz saqlash va foydalanuvchilar tomonidan foydalanishni ta'minlaydi.

Bu vazifalar tizimni o'rganish, o'qituvchilar va o'rganuvchilar uchun qulayliklar taqdim etish, o'qitish jarayonlarini avtomatlashtirish va imtihon natijalarini tahlil qilishda yordam beradi.

Kurs ishini optimallashtirish, o'quv jarayonini samarali va tushunarli qilishga yordam beradigan avtomatlashtirilgan axborot tizmini (AIT) ishlab chiqishning bir qismi hisoblanadi. Bu jarayon, o'quvchilarning bilimini oshirish, o'quv jarayonini tashkil etish, ma'lumotlarni to'plash va saqlash, natijalarni hisoblash va analiz qilish, shuningdek o'quvchilarning o'zgartirishlarini kuzatish va adaptiv o'quvni amalga oshirishga yordam beradi. AITning ishlab chiqilishi, o'quvchilarning o'zlarining o'quv jarayonini boshqarishga imkon beradi, shuningdek o'qituvchilarning va o'quvchilarning vaqtini samarali ishlatishga yordam beradi. Bu, o'quvchilarning bilimini oshirishga va o'quv jarayonini yaxshilashga yordam beradi, shuningdek o'quvchilarning o'zgartirishlarini kuzatish va adaptiv o'quvni amalga oshirishga yordam beradi.

Kurs ishini optimallashtirishning asosiy vazifalari quyidagilardan iborat:

Test Savollarini Tuzish: AIT, fan bo'yicha test savollarini avtomatlashtirib, o'quvchilarning bilimini tekshirish uchun zarur bo'lgan savollarni tuzadi. Bu jarayon, o'quvchilarning bilimini oshirishga yordam beradi.

Ma'lumotlarni To'plash va Saqlash: o'quvchilarning javoblarini, o'quvchilarning o'zgartirishlarini va boshqa ma'lumotlarni to'plash va saqlash uchun zarur bo'lgan texnologiyalarni ishlatadi.

Natijalarni Hisoblash va Analiz Qilish: AIT, o'quvchilarning javoblarini baholash va natijalarni hisoblash uchun zarur bo'lgan algoritmlarni ishlatadi. Bu, o'quvchilarning bilim darajasini oshirishga yordam beradi.

O'quvchilarning O'zlashtirishlarini Kuzatish: ya’ni, o'quvchilarning o'zlashtirishlarini kuzatish va ularning o'quv jarayonini yaxshilashga yordam beradi.Buni faqat admin ya’ni o`qituvchi kuzatishi mumkin.

**Kurs ishining tuzilishi:** Kurs ishi kirish qismi, mundarija, xulosa va ilmiy-amaliy tavsiyalar hamda foydalangan adabiyotlar ro‘yxatidan iborat.

Kurs ishining birinchi bobi avtomatlashtirilgan axborot tizimini ishlab chiqishda ishlatilgan dasturiy ta‘minotlarga tasnifi haqida. Birinchi bobda ikkita bo‘lim mavjud:

**Dasturlash Tillari va Texnologiyalar:** Dasturlash tillari, masalan, C#, Java, Python, PHP, ASP.NET, Django kabi, va dasturlash texnologiyalari, masalan, Visual Studio, PyCharm, Eclipse, IntelliJ IDEA, va boshqalari kiritiladi. Bu tillar va texnologiyalar tizimning yaratilishi va ishga tushirilishi jarayonlarida foydalaniladi.

**Ma'lumotlar Bazalari:** Ma'lumotlar bazalari tizimning asosiy qismi hisoblanadi. Ular, savollar, javoblar, foydalanuvchilar, imtihon natijalari, mavzular va boshqa ma'lumotlarni saqlash uchun xizmat qiladi. SQL Server, MySQL, SQLite, MongoDB kabi ma'lumotlar bazalari ko'p foydalaniladi.

Bu ta'minotlar, tizimni yaratish va ishga tushirish jarayonlarida foydalanilgan dasturiy va texnik ta'minotlarni tasniflashda yordam beradi. Har bir ta'minot o'zining o'ziga xos vazifalarni bajarishda muhim ahamiyatga ega.

Tizimni yaratishda qo`llanilgan dasturiy vositalar.

Avtomatlashtirilgan axborot tizmini yaratishda qo`llanilgan berilganlar bazasi haqida.

Kurs ishining ikkinchi bobida avtomatlashtirilgan axborot tizimini yaratilishi haqida to‘xtalib o‘tilgan. Ikkinchi bobda ikkita bo‘lim bo‘lib, Loyihaning yaratilish qismi bilan tanishish, Yaratilgan tizimining amaliy ahamiyati va qo’llanish sohalari.

# I-BOB. AVTOMATLASHTIRILGAN AXBOROT TIZIMINI ISHLAB CHIQISHNING NAZARIY ASOSLARI

## 1.1. Tizimni yaratishda qo`llanilgan texnik va dasturiy vositalar haqida malumot va ularning imkoniyatlari

C# Windows Forms, C# Windows Form yoki C# WinForms, Microsoft tomonidan taqdim etilgan grafik foydalanuvchi interfeysi (GUI) platformasi hisoblanadi. Ushbu platforma o'rnatilgan dasturlar yaratish uchun Windows asosli ilovalarni ishlab chiqishga imkoniyat beradi. Bu dastur ishlab chiquvchilarga yuqori sifatli va interaktiv foydalanuvchi interfeysli Windows asosli dasturlar yaratish imkoniyatini beradi. C# Windows Forms orqali dasturchilar odatda quyidagi vazifalarni bajarishlari mumkin:

C#, Microsoft tomonidan ishlab chiqilgan, .NET platformasida foydalanuvchilarga ilova ishlab chiqishda yordam beruvchi dasturlash tili hisoblanadi. Quyida C# haqida bilish mumkin:

Tarixi: C# 2000-yilning boshida Microsoft tomonidan ishlab chiqilgan bo'lib, .NET Framework, keyinchalik .NET Core va .NET 5+ kabi platformalarda ishlatiladi. Uning ismi "C sharp" deb o'qiladi va "C++" dasturlash tilining yangi versiyasi sifatida tasavvur qilingan.

Muqova: C#, asosan korporativ tizimlar, web ilovalari, o'yinlar, Windows ilovalari, mobil ilovalar, IoT (Internet of Things) va boshqa turlarda foydalaniladi. Microsoft platformalarida C# juda mashhurdir, lekin uning keng qo'llanilishini ko'rsatish uchun boshqa muhitlar ham mavjud.

Sintaksis: C#, C va C++ dasturlash tillari asosida yaratilgan, shuning uchun ular o'rtasida o'zaro o'xshashlik bor. C# esa oddiy va chuqur sintaksisga ega, shuningdek, yadro funktsiyalarni, yadro avtomatik boshqaruvni va boshqa zarur dasturlash qo'llanmalari bilan ta'minlanadi.

Platforma: C# Microsoft .NET (vaqtincha .NET Core va .NET 5+ hisoblanadi) platformasida ishlaydi. Bu, Windows, macOS, Linux, iOS, Android va boshqa muhitlar uchun yordamchi dasturlash platformalariga dastlab ta'minlashni maqsad qilgan.

Xususiyatlar: C#, ob'ektga yo'naltirilgan dasturlash tillari (OOP) prinsiplarini qo'llab-quvvatlaydi. U, asosan klaslar, interfeyslar, o'zaro aloqalar, nasl, polimorfizm, inkapsulatsiya, boshqaruv paneli va boshqalar kabi ob'ektga yo'naltirilgan qo'llanmalarni qo'llab-quvvatlaydi.

Kompaniyalar va Jamoalar: C# ni eng ko'p Microsoft tomonidan ishlatiladi, ammo boshqa kompaniyalar ham, masalan, Amazon, Facebook, Google, Apple va boshqalar, uning ishlatilishini oshirish uchun ishlatiladi.

Qo'llanma: Visual Studio, Visual Studio Code va Xamarin Studio kabi dasturlash muhitlari C# ni dasturlash va ishlab chiqishda yordam beradi.

C#, keng doiradagi sohalarda dasturlash uchun kuchli va samarali til sifatida taniladi. Uning oson o'rganish uchun ko'p darsliklar, onlayn resurslar va jamlanmalar mavjud.

Windows asosli ilovalarni ishlab chiqish: C# Windows Forms, Windows operatsion tizimida ishlaydigan ilovalarni yaratish uchun moslashtirilgan. Bu orqali, shaxsiy kompyuteringizda Windows ochiq dasturlarini ishlab chiqishingiz mumkin.

Foydalanuvchiga interaktiv interfeys berish: C# Windows Forms grafik komponentlarni taqdim etadi, masalan, tugmalar, ko'rinishlar, jadvallar va boshqalar. Bu komponentlar orqali foydalanuvchiga dasturning funktsiyalariga kirish, ma'lumotlarni kiritish va boshqa amallarni bajarish imkoniyatini beradi.

Sodda dastur interfeysi yaratish: C# Windows Forms dasturlarini yaratishda Visual Studio kabi yordamchi muharrir dasturlaridan foydalanish mumkin. Bu dasturlar interfeys elementlarini tasvir qilish va ularga funktsiyalarni bog'lash uchun intuitiv vositalar taqdim etadi.

Ushbu platforma o'rnatilgan dasturlar oddiy .NET Framework-yordamida ishlaydi va C# dasturlash tilidan foydalanishni talab qiladi. Ushbu dasturlash tilining yaxshi o'rganishga va oddiy sintaksisiga ega bo'lish, Windows ilovalari yaratishda ishlatilishi uchun katta afzallikdir.

C# WindowsForm (Windows form .Net) – bu UI framework bo‘lib kompyuter dasturlarini yaratishda foydalaniladi. Ishlab chiqish platformasi boshqaruvni boshqarish, grafikalar, maʼlumotlarni bog‘lash va foydalanuvchini kiritish kabi dasturlarni ishlab chiqish funksiyalarini o‘zida jamlagan. WinForm Visual Studio orqali formaga komponentlarni osongina tashlash va ularni qo‘zg‘atish orqali kerakli joyga joylashtirib visual dizayn yaratish mumkin.

Visual Studio dasturini ishga tushiramiz. “Create a new project” bo‘limi tanlanadi. Search for templates bo‘limiga winforms deb yozamiz va ENTER ni bosamiz. Dasturlash tillari orasidan C# ni tanlaymiz. Template(andoza)lar orasidan Windows Forms App (.NET Framework) ni tanlaymiz. Configure your new Project menyusi ochiladi va biz endi yaratmoqchi bo‘lgan dasturimiz nomini Project name bandiga yozamiz va Create tugmasini bosamiz.

Muhit bilan tanishish. Visual Studioda Windows Form ko‘rinishidagi dasturlarni yaratishda e’tiborimizni qaratishimiz zarur bo‘lgan asosiy 4 ta qism to‘g‘risida to‘xtalib o‘tamiz.

Solution Explorer Yaratilgan loyihaning qismlari, fayllari, kodlari, resurs(manba)lari shu oynada ko‘rinib turadi.

Properties (xususiyatlar) Bu oynda tanlangan elementning xususiyatlarini sozlash mumkin bo‘ladi. Misol uchun, Solution Explorer dan biror bir dastur qismi yoki faylni tanlasangiz ushbu qism uchun mavjud bo‘lgan xusuyitalarni sozlash mumkin bo‘ladi. Bundan tashqari Forma va unga tashlangan komponentlarni ham xususiyatlarini o‘zgartirish mumkin.

Form Designer (forma dizayn qismi) Formaning dizyan qismi bu yerga Toolbox dan komponentlarni tashlashingiz va joylashtirishingiz mumkin. Bu orqali foydalanuvchi uchun qulay va chiroyli bo‘lgan foydalanuvchi interfeysi (user interface - UI) tayyorlaysiz.

Toolbox Bu yerdan dasturingiz uchun kontroller(komponentlar) ni olishingiz mumkin. Buning uchun kerakli komponent ustiga ikki marta bosiladi yoki komponentni tanlab surib olib o`tsa ham bo`ladi.

SQL Server maʼlumotlar bazasi bilan ishlash. MS SQL Server kabi maʼlumotlar bazasidagi maʼlumotlar ustidagi asosiy operatsiyalarni ko‘rsatadigan dasturni ishlab chiqamiz, xususan:

• ilovaga maʼlumotlar bazasi ulanishi;

• maʼlumotlar bazasi jadvallarini formada ko‘rsatish;

• maʼlumotlar bazasiga yangi yozuv qo‘shish;

• yozuvni o‘chirish.

ORM(OBJECT RELATIONAL MAPPING)

Yangi texnologiyaning qadrini tushunish uchun undan noqulay, effektivligi past, konsepsiyasi boshqacha texnologiyalardan foydalanib ko'rish kerak. Bunga bir qancha tendensiyalarni keltirib o'tish mumkin. Misol uchun Obyektga yo'naltirilgan dasturlash konsepsiyasining muhumliligini tushunish va qadriga yetish uchun Funksional dasturlash konsepsiyasini tushunib olish kerak yoki aksincha.

Yaratilyotgan dastur ma'lumotlarni bir nechta formatlardan foydalanadigan manbalardan olishi mumkin. Bunga misol qilib CSV, JSON formatlari va turli databasetlarni misol keltirish mumkin.

Birinchi 2 ta texnikaning ishlash prinsipi faylga ma'lumotni ma'lum formatda yozish va o'qib olish edi. Lekin, o'qib olayotgan paytda dasturimiz uchun yozilgan ma'lumot biz kutgandek to'g'ridan-to'g'ri tushunarli emas. Endilikda bizning vazifamiz yozilgan ma'lumotni o'zimiz ishlayotgan platforma tushunadigan ma'lumot turiga aylantirishdir.

Ya'ni biz bu paytda ma'lum qoida asosida ma'lumotlarni obyektlarga aylantiradigan (convert) logika yozishimiz yoki shu logikani bajaradigan package o'rnatishimiz kerak (misol: Json serializer & deserializer). Bu narsani ma'lumot saqlaydigan manbamiz va dasturimiz orasidagi ko'prik desak bo'ladi.

Endi asosiy mavzuga qaytadigan bo'lsak ORM (OBJECT RELATIONAL MAPPING) bu OOP tillari va RELATIONAL database'lar orasidagi ana shunday ko'prikni bajaradigan, ikkovini ulab beradigan texnika.

Copy

.NET ---> ORM ---> Database

.NET <--- ORM <--- Database

OOP tillaridan foydalangan holda database bilan ishlashda biz undan ma'lumotlarni yaratish, o'qish, yangilash va o'chirish (CRUD) kabi turli xil vazifalarni bajarishimiz kerak bo'ladi. O'zi aslida, bu narsalarni bajarish uchun SQL'dan foydalanishimiz kerak edi va SQL shu uchun ham yaratilgan. Lekin development paytida fokus ikkiga bo'linmasligi uchun ishimizni osonlashtirib berish uchun ORM turli qulayliklarni bizga taqdim etadi. Endilikda qiynalib SQL so'rovlari orqali ma'lumot ustida amal bajarishdan ko'ra xuddi shu so'rovlarning vazifasini bajaradigan lekin bizga sintaksisi qulay bo'lgan metodlardan foydalanishimiz mumkin.

.NET ekosistemasida bir qancha ORM texnologiyalari mavjud. Ulardan mashhurlarini quyida bilib olasiz:

Entity Framework;

NHibernate;

Dapper;

BFC

Bulardan tashqari boshqa texnologiyalar ham mavjud va bulardan aynan bittasini kuchli deyish noto'g'ri, chunki texnologiyalar project'ga qulay qilib tanlanadi va har xil project uchun har xil texnologiya mos kelishi mumkin.

ORMning foydali tomonlari:

* Dasturning development vaqtini qisqartiradi;
* Database bilan ishlashdagi mantiqni boshqarishni osonlashtiradi;
* Kamroq va osonroq kod orqali ish bitadi;

ORMning kamchiliklari:

* SQLdan ko'ra sekinroq;
* Kattaroq va murakkabroq so'rovlar bilan ishlashda kamchiliklari seziladi;

Ma'lumot o'rnida muhim bir narsani aytib o'tish kerakki, ORMni o'rganish sizni hech qachon SQLni o'rganishdan to'xtatishi kerak emas. ORM shunchaki sizning ishingizni osonlashtirish uchun yaratilgan. ORM yaratgan qulayliklarni SQLni tushunmasdan o'rganyotgan bo'lsangiz avvalo uni o'rganishingizni tavsiya beraman.

**.NET**

.NET Framework — bu kodning bajarilishini boshqaruvchi Common Language Runtime (CLR) va ilovalarni yaratish uchun sinflarning boy kutubxonasini ta’minlovchi Base Class Library (BCL) larni o’z ichiga olgan ishlab chiqish platformasi.

Zamonaviy operatsion tizimlar va dasturlar 64 Gigabayt .NET Framework — 2002-yilda Microsoft tomonidan chiqarilgan dasturiy platformadir. Platforma turli dasturlash tillari: C#, Visual Basic .NET va boshqalar uchun mos Common Language Runtime (CLR)ga asoslangan. CLR funksiyasi ushbu platformadan foydalanadigan har qanday dasturlash tilida mavjud. .NET Framework hozirda .NET sifatida rivojlanmoqda. Bu platformada koʻp dasturlarga umumiy komponentlar va optimizatsiyalangan metodlar bor.

.NET Framework oʻsha paytda mashhur boʻlgan Sun Microsystems (hozirda Oracle kompaniyasiga tegishli) Java platformasiga Microsoftning javobidir.

.NET Framework Microsoft kompaniyasining oʻz mahsuloti hisoblanib, rasmiy ravishda Windows operatsion tizimlarida ishlash uchun moʻljallangan boʻlsa-da, baʼzi boshqa operatsion tizimlarda .NET Framework dasturlarini ishga tushirish imkonini beruvchi mustaqil loyihalar (birinchi navbatda Mono va Portable.NET) mavjud.

Platformani ishlab chiqish 1999-yilda boshlangan. Rasmiy ravishda, yangi texnologiyaning ishlab chiqilishi 2000-yil 13-yanvarda, Bill Geyts Microsoft rahbarligining Stiv Balmerga oʻtkazilishini paytida rasman eʼlon qilindi. Shu kuni korporatsiya rahbariyati kompaniya uchun Next Generation Windows Services (NGWS, “Windows xizmatlarining yangi avlodi”) deb nomlangan yangi strategiyasini eʼlon qildi. Yangi strategiya foydalanuvchilarga Internetga kirish imkoniga ega simsiz qurilmalardan shaxsiy kompyuteri (PC)da boʻlgani kabi Internet bilan ishlash imkonini berish uchun mavjud va kelajakdagi Microsoft ishlanmalarining yagona toʻplamiga birlashtirish edi.

Oʻsha kuni boʻlib oʻtgan matbuot anjumanida Balmer shaxsiy kompyuterlarining ulkan imkoniyatiga qaramay, korporatsiya keyingi avlod xizmatlarining shaxsiy kompyuter boʻlmagan qurilmalarda ham ishlashi kafolatlanishini taʼminlash muhimligiga urgʻru bergan. Geytsga kelsak, Patricia Seybold Group tahlilchisi Enn Tomas Meyns uning soʻzlarini tahlil qilib, kompaniyaga faqat ish kompyuterlariga eʼtibor qaratishdan voz kechib, mobil qurilmalarni ishlab chiqishga oʻtish foyda keltirishini aytdi. Uning fikriga koʻra, Geyts Windows bilan qattiq bogʻlanmagan kompyuterda ishlashning yangi davriga yoʻl ochdi.

Mobil qurilmalarning xotira hajmi kichik boʻlganligi tufayli ilovalarni saqlash va uzatish serverlar tomonidan amalga oshirilishi lozim deb qaralgan, holbuki oʻsha paytda deyarli barcha foydalanuvchi maʼlumotlari va dasturiy taʼminot statsionar kompyuterlarda lokal ravishda saqlangan. Keyin “serverga asoslangan” modelga oʻtish gʻoyasi eng yirik IT kompaniyalar rahbarlari orasida kuchli qoʻllab-quvvatlandi. Misol uchun, Sun Microsystems rahbari Skott Maknili server-mijoz dasturiy taʼminotiga oʻtishda Microsoft Office bilan raqobatlashadigan ofis dasturiy taʼminoti kompaniyasini sotib olganini eʼlon qilgan.

**NET arxitekturasi**

Har qanday qoʻllab-quvvatlanadigan dasturlash tilida yozilgan .NET Framework dasturi birinchi navbatda kompilyator tomonidan .NET Common Intermediate Language (CIL) (ilgari Microsoft Intermediate Language, MSIL) bayt-kodiga tarjima qilinadi. .NET nuqtayi nazaridan, bu yigʻish (assembly) deb hisoblanadi. Keyin kod Common Language Runtime (CLR) virtual mashinasi tomonidan bajariladi yoki NGen.exe tomonidan maʼlum maqsadli protsessor uchun bajariladigan kodga tarjima qilinadi.

Virtual mashina (VM)dan foydalanish afzalroq, chunki u ishlab chiquvchilarni apparatning oʻziga xos xususiyatlari haqida tashvishlanishdan xalos qiladi. CLR virtual mashinasidan foydalanilganda, oʻrnatilgan JIT-kompilyator ”shu ondayoq” (just in time) oraliq baytkodni kerakli protsessorning mashina kodlariga aylantiradi. Dinamik kompilyatsiya qilishning zamonaviy texnologiyasi yuqori darajadagi ishlashga erishishga imkon beradi. CLR VM shuningdek, asosiy xavfsizlik, xotira boshqaruvi va istisnolar bilan shugʻullanadi, bu esa ishlab chiquvchiga ishning bir qismini tejaydi.

.NET Framework arxitekturasi Microsoft tomonidan ishlab chiqilgan va ISO va ECMA tomonidan tasdiqlangan Common Language Infrastructure (CLI) spetsifikatsiyasida tavsiflangan va ishlab chiqilgan. CLI .NET maʼlumotlar turlarini, dastur tuzilmasi metamaʼlumotlar formatini, bayt kodini bajarish tizimini va boshqalarni tavsiflaydi.

Barcha qoʻllab-quvvatlanadigan dasturlash tillari uchun mavjud boʻlgan .NET obyekt sinflari Framework Class Library (FCL)da mavjud. FCL tarkibiga Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation va boshqalar kiradi. FCLʼning yadrosi asosiy sinf kutubxonasi (BCL) deb ataladi.

.NET platformasini quyidagi dasturlar qoʻllab-quvvatlaydi:

• Microsoft Visual Studio (C#, Visual Basic .NET, Managed C++, F#);

• SharpDevelop;

• MonoDevelop;

• Embarcadero RAD Studio (Delphi for .NET); oldingi Borland Developer Studio (Delphi for .NET, C#);

• A# (Ada);

• Zonnon;

• PascalABC.NET;

• JetBrains Rider.

.NET ilovalari matn muharririda kompilyatorni buyruqlar satridan oddiygina chaqirish orqali ham foydalanish mumkin.

Microsoft .NETʼning asosiy gʻoyalaridan biri bu turli tillarda yozilgan dasturiy qismlarning mosligidir. Masalan, Microsoft .NET uchun C++ tilida yozilgan xizmat Delphida yozilgan kutubxonadan sinf usuliga kirishi mumkin; C#ʼda siz Visual Basic .NETʼda yozilgan sinfdan meros boʻladigan sinfni yozishingiz mumkin va C#ʼda yozilgan usul bilan chiqarilgan istisno Delphida amalga oshirilib, ishlov berilishi mumkin. .NETʼdagi har bir kutubxonada (montajda) oʻz versiyasi haqida maʼlumot mavjud boʻlib, bu toʻplamlarning turli versiyalari oʻrtasida yuzaga kelishi mumkin boʻlgan ziddiyatlarni bartaraf etish imkonini beradi.

Microsoft Visual Studio bilan taʼminlangan tillar:

• C#

• Visual Basic .NET

• JScript .NET

• C++/CLI — Managed C ++ ning yangi versiyasi

**ADO.NET**

ADO.NET bu shunday sinflar to’plamiki, u C# va .NET Frameworkda relyatsion jadvallar va ma’lumotlar bilan ishlashda qulaylik yaratadi. Relyatsion ma’lumotlar bazalari bo’lmish Microsoft SQL Server va Microsoft Access bilan bir qatorda boshqa relyatsion ma’lumotlar bazalari va hattoki relyatsion bolmagan ma’lumotlar manbalarni ham o’z ichiga oladi. ADO.NET texnologiyasi barcha .NET dan foydalanuvchi dasturlash tillarida .NET Framework va loyihalashni yaxlid holda birlashtirishda foydalaniladi, u C#ning o’ziga xos xususiyati hisoblanadi. ADO.NET System.Data sinfi va uning sinf ostilari bo’lmish System.Data.SqlClient va System.Data.Linq bilan bir qarorda System.Xml sinfi va sinf ostilarining bir qismini oz ichiga olgan sinflardan tashkil topgan.

Nimaga u ADO.NET deb nomlandi?. Balki siz o’ylayotgandirsiz nima uchun ADO.NET bo’limini .NET Framework o’z ichiga oldi? Nima uchun uni shunchaki System.Data va boshqa nom bilan nomlanmadi? ADO.NET ActiveX Data Objects (ADO) ning kengaytirilgan sinflar to'plami bo’lib, u ma’lumotlardan Microsoftning oldingi avlod texnologiyasiga nisbatan tezroq foydalanish imkonini beradi. ADO.NET- ADO ga nisbatan ma’lumotlardan tez foydalanuvchi sinflar va .NET dasturlash muhitlarini yangilash va kengaytirishda xizmat qildi. Agarda sizni bu texnologiyaning yaratilish tarixi, boshqa interfeysdan turib Microsoft ma’lumotlar bazasi bilan ishlash qanday amalga oshirilishi, shunga o’xshash ODBC yoki OLE DB ishlashi qiziqtirsa, pastda keltirilgan qisqacha tarixga ko’z tashlang. Ma’lumotlar bilan ishlash haqida qisqacha tarix. Qachonlardir malumotlarni boshqarish tizimi birinchi bo’lib yaratildi, bular Oracle va DB2 IBM uchun, istalgan dasturchiga qaysidir darajada ma’lumotlardan foydalanishi zarur, ular har bir malumotlni boshqarish tizimlari uchun funksiya yaratishga ehtiyoj sezdilar. Har qaysi tizimlar uchun o’zining funksiyalar kutubxonasi mavjud bo’lib, bular Oracle uchun Oracle Call Interface (OCI), Sybase SQL Server (keyinchalik uni Microsoft sotib olgan) uchun DBLib. Dasturlarning ma’lumotlardan juda tez foydalanish va ular ma’lumotlar bazalari bilan to’g’ridan to’g’ri bog’lanish imkonini beradi. Ammo dasturchi har bir ma’lumotlar bazalari uchun ularning kutubxonalarini bilishi, ularda ishlab biror natija ola bilishi zarur edi. Shu sababli ma’lumotlarni boshqarish tizimlaridan foydalanib dastur yozish haddan tashqari murakkab masala edi. Bu shundan dalolat beradiki, agar shirkat(daturiy mahsulotlardan foydalanuvchi) oldingi ma’lumotlar bazasi o’rniga bosqasidan foydalanmoqchi bo’lsa u barcha ilova dasturlarini(приложения) qaytadan yozishiga to’g’ri kelar edi. Bu muammo ODBC (Open Database Connectivity ) yaratilishi bilan o’z yechimini topdi.ODBC ustida Microsoft boshqa shirkatlar bilan birgalikda 90- yildan boshlab ishlay boshladi. ODBC dasturlovchilarga barcha funksiyalari yordamida xoxlagan ma’lumotlar bazasi tizimidan foydalanish imkonini beradi. Bu funksiyalar ma’lumotlar bazasidan foydalanayotganda aynan shu ma’lumotlar bazasini boshqarish tizimining drayverlarida translatsiya qilinadi. Bu orqali koplab muammolar yechildi, patentlangan ma’lumotlar baza kutubxonalari bilan islovchi dasturchilar endi bir turdagi funksiyalar to’plamidan foydalanishadi(ODBC funksiyalaridan); agarda shirkat o’zining ma’lumotlar bazasini boshqarish tizimini o’zgartirsa ular faqat kodning ma’lumotlar bazasini bog’lovchi qisminigina o’zqartirishadi xolos. Keyinchalik yana bir muammo paydo bo’ldi. Bir qancha ofeslar shirkat ishini bir joydan turib katta hajmdagi ma’lumotlarni qog’ozlarsiz elektron ko’rinishda saqlash haqida orzu qila boshladilar yani elektron kutubxona, web sahifalar, Project 2000 fayllari va boshqalar. ODBC an’anaviy ma’lumotlar bazalari bilan juda yaxshiishlaydi, lekin ma’lumotlar tipini birinchisidan boshqasiga o’tkazishm satr bo’yicha tartiblash va ustun bo’yicha jolashtirishda bazida umuman asosiy tuzilish bilan mutanosiblikni yo’qotardi. OLE DB paydo bo’lganidan so’ng bu muammo o’z yechimini topdi. OLE DB texnologiyasining ishlash usuli ODBCga o’xshash, ma’lumotlar bazasidan ma’lumotlar abstrak qatlam orqali ilova dasturga beriladi, faqat ruxsat berilgan ma’lumotlargina. Foydalanuvchi ilova dasturi ma’lumotlar manbasiga(источник) an’anaviy ma’lumotlar bazasi yoki xohlagan joyda saqlanayotgan ma’lumotlar OLE DB uzatuvchisi orqali yetkaziladi. Ma’lumotlar ilova dasturga jadval shaklida uzatiladi — bunda ular xuddi ma’lumotlar bazasi bilan ishlayotganday bo’ladi. OLE DB ma’lumotlarni uzatishda ODBC drayverlaridan foydalar, balki ma’lumotlar ba’zasidan foydalanishda ODBCni qo’llar. Keyingi bo’limlarda ADO.NET OLE DB va ODBCning ustida qurilgan va ularga juda o’xshash ekanligiga ishonch hosil qilasiz. Ma’lumotlar bilan ishlashning keyingi avlod texnikasi .NET davri paydo bo’lishi bilan ActiveX Data Objects (ADO) yaratildi. ADO OLE DBdan yuqori tabaqa bo’lib OLE DBning ustiga qurilgan. Yuqori bosqichli dasturlash tillarida dastur yozishda Visual Basic da OLE DBdan foydalanib ma’lumotlarga ishlov beriladi. ADO.NET texnologiyasi .NET Framework 1.0 bilan bir vaqtda yaratildi va .NET har xil versiyalari uchun ADO.NETning ma’lumotni yetkazuvchilari qo’shilib rivojlantirildi, sekin asta mukammallashtirildi. NET Framework paydo bo’lishi bilan LINQ to SQL o’zining birinchi ma’lumotlar bilan ishlash uchun muhim ―siljish paradigmlari‖ni namoyish qildi. ADO.NET endigi rivojlanishni LINQ bilan birgasi takominlashtirdi. Kutilmaganda yangi ADO.NET for Entities mahsulot yaratildi u Visual Studio 2008 dan boshlab ishlatila boshlandi.

* ADO.NET loyihalashning maqsadi. Quyida ADO.NET loyihalashning asosiy maqsadlarini qisqacha keltiramiz.
* Relyatsiyon va relyatsiyon bo’lmagan malumotlar bilan oddiy bog’lanish.
* Oldingi texnologiyalarga nisbatan koproq ma’lumotlarni qamrab olishi va keng qamrovli qo’llanilishi.
* Internet uchun yuqori darajadagi ilova dasturlarda qo’llash.
* XML ma’lumotlar va relyatsion ma’lumotlardan bir xil foydalanish.

Bugungi kunda ma’lumotlar bilan ishlash katta ahamiyatga ega. Ma’lumotni saqlash uchun turli xil ma’lumotlar bazalarini boshqarish tizimlari qoʻllaniladi: MS SQL Server, Oracle, MySQL va boshqalar. Koʻpgina katta dasturlar, bu yoki boshqa usulda, ma’lumotlarni saqlash uchun ushbu ma’lumotlar bazasini boshqarish tizimlaridan foydalanadilar. Ammo ma’lumotlar bazasi va C # ilovasi oʻrtasida aloqa oʻrnatish uchun vositachi kerak. Va aynan shunday vositachi ADO.NET texnologiyasidir.

ADO.NET .NET Framework asosida ma’lumotlar texnologiyasini ta’minlaydi. Ushbu texnologiya bizga ma’lumotlar bazalariga soʻrovlar yuborish, ulanishlarni oʻrnatish, ma’lumotlar bazasidan javob olish va boshqa bir qator operatsiyalarni bajarish uchun sinflar toʻplamini taqdim etadi.

Bundan tashqari, ma’lumotlar bazasini boshqarish tizimlari juda koʻp boʻlishi mumkinligini ta’kidlash kerak. Oʻzlarining mohiyatiga koʻra, ular farq qilishi mumkin. Masalan, SQL Server soʻrovlarni tuzishda T-SQL dan foydalanadi, MySQL va Oracle esa PL-SQL-dan foydalanadilar. Turli ma’lumotlar bazalari tizimlarida har xil turdagi ma’lumotlar boʻlishi mumkin. Boshqa fikrlar ham farq qilishi mumkin. Biroq, ADO.NETning funktsional imkoniyatlari ishlab chiquvchilarga keng qamrovli DBMS bilan ishlash uchun birlashtirilgan interfeysni taqdim etadigan tarzda yaratilgan.

Microsoft dastlab .NET Frameworkni kross-platform imkoniyatiga ega bo’lishini rejalashtirgandi, biroq Microsoft uni Windows bilan juda zo’r ishlashini amalga oshirish uchun kuchini sarfladi.

.NET Framework 4.5.2 dan boshlab u Windows operatsion tizimining rasmiy komponenti hisoblanadi.

Komponentlar o’zlarining asosiy mahsulotlari bilan bir xil qo’llab-quvvatlovga ega, shuning uchun 4.5.2 va undan keyingilari o’rnatilgan Windows operatsion tizimning yashash davri siyosatiga amal qiladi. .NET Framework bir milliarddan ziyod kompyuterlarda o’rnatildi, shuning uchun unda iloji boricha kamroq o’zgarishlar bo’lishi kerak. Hatto bug(qurtcha) larni tuzatish ham muammolarga olib kelishi mumkin, shunga u kamdan-kam yangilanadi.

GUNA va NuGet paketlarining farqlarini va ularning bir-biriga qanday qo'llanildiğini tushunish uchun, ularni alohida ko'rib chiqamiz

GUNA, Grafikali Foydalanuvchi Interfeysi (GUI) komponentlarini ta'minlovchi bir kutubxona yoki paketdir. Bu paket, Windows Forms ilovasining dizaynini chiroyli va qulay qilish uchun ishlatiladi. GUNA komponentalari, boshqa standart Windows Forms elementlaridan farqli ravishda, ko'proq dizayn imkoniyatlari va animatsiyalarga ega. Ular, dastur interfeysini estetik va intuitiv qilishga yordam beradi. GUNA komponentalari dastur ishlab chiqishda va interfeys dizaynida qulaylik yaratish uchun ishlatiladi.

Guna paketi, Windows Forms ilovasining dizaynini chiroyli va qulay qilish uchun ishlatiladi. Ushbu paket, GunaUI kompaniyasi tomonidan yaratilgan. Bu kompaniya Windows Forms ilovalarini ishlab chiqish va ularga chiroyli dizayn qo'shishda mutaxassis bo'lgan dasturchilar tomonidan olib borilgan.

Guna paketi, Guna komponentalari deb nomlangan dizayn elementlarini o'z ichiga oladi. Ular, standart Windows Forms elementlaridan farqli ravishda, dizayn imkoniyatlari va animatsiyalarga ega bo'lib, dasturlarni chiroyli va samarali qilishda yordam beradi. Bu komponentlar, Visual Studio muhitida dastur yaratish jarayonida qulay va yaxshi ko'rinishli interfeys yaratish uchun ishlatiladi.

Guna paketining tarixi:

GunaUI kompaniyasi, Guna paketini dastur yaratish sohasidagi ehtiyojlarini aniqlab, dasturchilar uchun chiroyli va samarali dizayn elementlarini taqdim etish maqsadida yaratgan. Ular, dasturchilar uchun qulayliklar va chiroylikni oshirishga qaratilgan Guna komponentlarini yaratishda mutaxassislar.

Universitetlar, kompaniyalar va shaxsiy loyihalar kabi ko'p yo'nalishlarda Guna paketidan foydalaniladi. Bu paket, interfeysni chiroyli va modern qilish, foydalanishning osonligi va ko'proq imkoniyatlar taqdim etishda yordam beradi.

Guna to'plami, xususan Guna UI Framework, ishlab chiquvchilar Windows ish stoli ilovalari uchun zamonaviy va vizual jozibador foydalanuvchi interfeyslarini yaratish uchun foydalanishi mumkin bo'lgan keng ko'lamli vizual komponentlarni taklif etadi. Guna to'plami tomonidan taqdim etilgan asosiy vizual komponentlardan ba'zilari:

Guna2Button: Turli holatlar, animatsiyalar va uslublarni qo'llab-quvvatlaydigan sozlanishi tugma boshqaruvi.

Guna2TextBox: To'ldiruvchi matn, ikonografiya, tekshirish va sozlash imkoniyatlarini qo'llab-quvvatlaydigan ilg'or matn kiritish boshqaruvi.

Guna2ComboBox: Shaxsiy elementlar, piktogrammalar va uslublarni qo'llab-quvvatlaydigan ochiladigan ro'yxat boshqaruvi.

Guna2DataGridView: Mavzular, uslublar va qo'shimcha funktsiyalarni qo'llab-quvvatlaydigan jadval ma'lumotlarini ko'rsatish va tahrirlash uchun kengaytirilgan ma'lumotlar tarmog'i boshqaruvi.

Guna2ToggleSwitch: Vizual fikr-mulohaza bilan variantlarni yoqish yoki o'chirish uchun zamonaviy almashtirish tugmasi boshqaruvi.

Guna2ProgressBar: Turli xil uslublar, animatsiyalar va vizual effektlarni qo'llab-quvvatlaydigan sozlanishi mumkin bo'lgan taraqqiyot panelini boshqarish.

Guna2RadioButton va Guna2CheckBox: Mavzular, uslublar va vizual sozlashni qo'llab-quvvatlaydigan sozlanishi mumkin bo'lgan radio tugma va belgilash katakchalari.

Guna2Panel: foydalanuvchi interfeysida boshqa boshqaruv elementlarini tashkil qilish va guruhlash uchun ko'p qirrali konteyner boshqaruvi.

Guna2DragControl: Shakllar yoki panellar kabi UI elementlarida sudrab olinadigan harakatni yoqish uchun yordamchi dastur komponenti.

Guna2PictureBox: o'lchamini o'zgartirish, kattalashtirish va turli xil displey rejimlarini qo'llab-quvvatlaydigan tasvirlarni ko'rsatish uchun boshqaruv.

Guna2NotificationControl: foydalanuvchilarga nojo'ya xabarlar yoki ogohlantirishlarni ko'rsatish uchun bildirishnoma komponenti.

Bular Guna to'plami tomonidan taqdim etilgan vizual komponentlarning bir nechta misollari. Umuman olganda, paket ishlab chiquvchilarga Windows ish stoli ilovalari uchun zamonaviy, sezgir va vizual jozibador foydalanuvchi interfeyslarini yaratish uchun boy vositalar to'plamini taqdim etadi.

Guna paketidagi vizual bo'lmagan komponentlar odatda UI elementlarini to'g'ridan-to'g'ri ko'rsatmaydigan, lekin vizual komponentlar uchun funksionallik yoki qo'llab-quvvatlashni ta'minlaydigan elementlarga ishora qiladi. Guna UI Framework yoki paketi kontekstida ba'zi umumiy novizual komponentlar quyidagilarni o'z ichiga olishi mumkin:

Sozlamalar: Ushbu komponentlar ishlab chiquvchilarga standart ranglar, shriftlar, uslublar va boshqalar kabi Guna UI elementlari uchun global sozlamalarni sozlash imkonini beradi.

Ma'lumotlarni ulash komponentlari: UI elementlari va asosiy ma'lumotlar modellari yoki manbalari o'rtasida ma'lumotlarni bog'lashni osonlashtiradigan komponentlar.

Voqealar ishlovchilari: UI o'zaro ta'siri uchun hodisalarni boshqarishni boshqaradigan komponentlar yoki sinflar, masalan, bosish hodisalari, hover hodisalari va boshqalar.

Animatsiya boshqaruvchilari: UI ichidagi animatsiyalar va o'tishlarni boshqarish, foydalanuvchi tajribasini yaxshilash uchun mas'ul bo'lgan komponentlar.

Layout Managers: Vizual bo'lmagan komponentlar, konteynerlar yoki shakllar ichidagi vizual komponentlar tartibini tartibga solish va boshqarishga yordam beradi.

Resurs menejerlari: UI ichida ishlatiladigan tasvirlar, piktogrammalar yoki lokalizatsiya fayllari kabi tashqi resurslarni yuklash, keshlash va boshqarish bilan shug'ullanadigan komponentlar.

Mavzu menejerlari: foydalanuvchilarga umumiy ko'rinish va his-tuyg'ularni dinamik ravishda o'zgartirishga imkon beruvchi ilovaning mavzuni almashtirish yoki sozlashni boshqaradigan komponentlar.

GunaElipse komponenti Guna UI Frameworkning bir qismi bo'lib, C# yordamida Windows ish stoli ilovalarini ishlab chiqish uchun keng qo'llaniladi. GunaElipse komponentining asosiy maqsadi boshqa boshqaruv elementlari yoki shakllarning burchaklarini yumalab, ularga zamonaviy va estetik jihatdan yoqimli ko'rinish berish qobiliyatini ta'minlashdir.

GunaElipse komponentisi ishlashi:

Komponentni qo'shing: Birinchidan, siz GunaElipse komponentini Visual Studio Designer-da Windows Form yoki UserControl-ga qo'shasiz.Keyin GunaElipse komponentining TargetControl xususiyatini o'rnatish orqali elliptik shaklni qaysi boshqaruv yoki shaklga qo'llashni belgilaysiz. Buni kodda yoki Visual Studio Dizayneridagi xususiyatlar oynasi orqali amalga oshirish mumkin.

Xususiyatlarni sozlash: Ixtiyoriy ravishda GunaElipse komponentining qo‘shimcha xususiyatlarini, masalan, ellips radiusi yoki effektni bolalar boshqaruviga qo‘llashni sozlashingiz mumkin.

Ushbu vizual bo'lmagan komponentlar Guna kabi mustahkam va xususiyatlarga boy UI tizimini ta'minlash uchun juda muhimdir, chunki ular dastur harakati va sahna ortidagi o'zaro ta'sirning turli jihatlarini ko'rib chiqadi, ishlab chiqish tajribasini va oxirgi foydalanuvchi qoniqishini oshiradi.

NuGet Package:

NuGet, .NET dasturlashida kutubxona boshqarish uchun muhim bir vosita hisoblanadi. Bu, kutubxonadagi paketlarni ishlab chiqish, yuklash, yangilash va boshqarish imkoniyatlarini ta'minlaydi. Dasturchilar, dastur yaratishda kerakli funksiyalarni qo'shish uchun NuGet paketlaridan foydalanishadi. NuGet paketlarini Visual Studio muhitida o'rnatish oson va samarali bo'lib, ularni dasturining funktsionalitetini kengaytirish uchun ishlatishadi.

NuGet, Microsoft tomonidan ishlab chiqilgan va ma'lumotlar yig'ilish vositasidagi o'lchamli dasturlar va keng ko'lamli ilovalar uchun ommaviy paket boshqaruvchi tizimi sifatida 2010-yilning yakunlarida e'lon qilindi. Ushbu platforma, .NET ilovalarni yaratish va ularga paketlar qo'shishni osonlashtirish maqsadida ishlab chiqilgan.

Qayerlarda qo'llanilishi:

NuGet paketlari, .NET dasturlashining umumiy platformasi sifatida ommaviy kutubxonalarda, loyihalarda, kompaniyalarda va Microsoft platformalarida juda keng qo'llaniladi. Visual Studio dasturlash muhitida, dasturlash jamoalarining loyihalarida, kompaniyalarda dasturlash va loyihalarni yaratishda va boshqa .NET dasturlarida NuGet paketlari juda keng qo'llaniladi.

Bu paketlar, .NET ilovalarida kerak bo'lgan ko'plab qo'shimcha funksiyalarni taqdim etish, xizmat qo'llab-quvvatlash, boshqa kutubxonalardan funksiyalarni olish va ko'proq imkoniyatlar bilan yaratish imkonini ta'minlashda juda yaxshi ishlaydi. NuGet paketlari, dasturchilar uchun kodni qisqartirish, qo'llab-quvvatlash va dasturlarni rivojlantirishda katta o'rinda qo'llaniladi.

Bundan kelib chiqib, GUNA paketi, GUI komponentlarini ta'minlovchi bir kutubxona bo'lib, dastur interfeysini chiroyli va intuitiv qilish uchun ishlatiladi, va uni o'rnatish uchun NuGet paketlaridan foydalanish mumkin. NuGet paketlarining yordamida GUNA komponentalari o'rnatilishi va yangilashlari oson va samarali bo'lib, dastur yaratishda dizayn va funktsionalitetni kengaytirish uchun yordam beradi.

## 1.2. Tizim uchun foydalanilgan berilganlar bazasi haqida

Agar berilganlar bazasi bo‘lmaganida ular axborot oqimi ichida qolib ketar edilar. Hozirgi kunda dasturlash tillar va dasturlash texnologiyalari haqida so‘z borar ekan. Har qanday axborot tizimining maqsadi real muhit ob‘ektlari haqidagi ma‘lumotlarga ishlov berishdan iborat. Keng ma‘noda ma‘lumotlar bazasi -bu qandaydir bir predmet sohasidagi real muhitning aniq ob‘ektlari haqidagi ma‘lumotlar to‘plamidir. Predmet sohasi deganda avtomatlashtirilgan boshqarishni tashkil qilish uchun o‘rganilayotgan real muhitning ma‘lum bir qismi tushiniladi. Masalan, korxona, zavod, ilmiy tekshirish instituti, oliy o‘quv yurti va boshqalar. Ma‘lumotlar bazasini yaratishda, foydalanuvchi axborotlarni turli belgilar bo‘yicha tartiblashga va ixtiyoriy belgilar birikmasi bilan tanlanmani tez olishga intiladi.

Ma’lumotlar bazasi – EHM xotirasiga yozilgan ma‘lum bir strukturaga ega, o‘zaro bog‘langan va tartiblangan ma‘lumotlar majmuasi bo‘lib, u biror bir ob‘ektning xususiyatini, holatini yoki ob‘ektlar o‘rtasidagi munosabatni ma‘lum ma‘noda ifodalaydi. Ma‘lumotlar bazasi tushunchasi maydon, yozuv, fayl (jadval) kabi elementlar bilan chambarchas bog‘liq

Ma'lumotlar bazasi (inglizcha: Database) markazlashtirilgan ma'lumotlar ombori hisoblanadi. U ma'lumotlarni qayta ishlash, qidirish, saqlash va o'qish uchun mo'ljallangan. Ma'lumotlar bazasi, dasturlash sohasida keng qo'llaniladi va muhim bo'lib, foydalanuvchilar ma'lumotlarni qayta ishlab chiqish, tahlil qilish va ma'lumotlar orasida aloqani amalga oshirishga imkoniyat beradi.

Quyidagi yetakchi ma'lumotlar bazalari mavjud:

Oracle Database: Oracle, eng mashhur va kuchli ma'lumotlar bazalari orasida hisoblanadi. U korporativ tizimlar, banklar, telekommunikatsiya kompaniyalari va boshqa katta tashkilotlar uchun yaxshi variant hisoblanadi. Oracle, bozorda tajribali va sertifikatsiyalangan dasturchilar tomonidan ommalashtirilgan.

Microsoft SQL Server: Microsoft SQL Server, Microsoft tomonidan ishlab chiqilgan va Windows tarmoqlari bilan integratsiya qilinadi. U keng doiradagi ilovalar va korporativ tizimlar uchun mo'ljallangan. SQL Server, tahlil va saxovatli ma'lumotlarni tahrirlash, o'qish va yozish imkonini beradi.

MySQL: MySQL, ochiq manba ma'lumotlar bazasi tizimi hisoblanadi. U eng keng tarqalgan ma'lumotlar bazalaridan biridir va katta tizimlarda, shuningdek, kichik va o'rta hajmdagi ilovalarda qo'llaniladi. Bunday tizimlar, veb-saytlar, bloglar, tijorat tizimlari va boshqa ilovalar uchun qulay va tez ishlash imkoniyatini ta'minlaydi.

Bu ma'lumotlar bazalari odatda SQL (Structured Query Language) deb nomlanadigan standart so'rovnoma tili orqali boshqariladi. Ushbu tillar, ma'lumotlarni saqlash, o'qish, yozish va boshqa amallarni bajarish uchun ishlatiladi. Har bir ma'lumotlar bazasi, uning tezligi, xavfsizligi, qo'llanilish qulayligi va ko'plab boshqa faktorlarga qarab tanlanadi.

Microsoft SQL Server, Microsoft tomonidan ishlab chiqilgan, kuchli va qulay ma'lumotlar bazasi tizimi hisoblanadi. Bu tizim, korporativ tizimlar, veb ilovalar, mobil ilovalar, analitika ilovalari, ma'lumotlar integratsiyasi va boshqa ko'plab sohalarda qo'llaniladi.

Qo'llanish: SQL Server, ma'lumotlar saqlash va boshqarish uchun oson foydalanish, bog'liq dasturlarni yaratish va ishga tushirish imkonini beradi. U ma'lumotlarni saqlash, o'qish, yozish, ma'lumotlar bazalarini boshqarish, avtomatik ruxsat boshqaruvini qo'llab-quvvatlash va boshqa kerakli funktsiyalarni ta'minlaydi.

Integratsiya: SQL Server, boshqa Microsoft mahsulotlari (masalan, .NET Framework, Visual Studio) bilan integratsiya qilinadi. Uning o'ziga xos manbalar va komponentlar mavjud bo'lib, tarmoq ma'lumotlar bazalariga (ODBC, JDBC kabi) kirish, tashqi ma'lumotlar manbalariga bog'lanish va integratsiya qilish imkoniyatini ta'minlaydi.

Xavfsizlik: SQL Server, ma'lumotlarni himoya qilish va xavfsizlikni ta'minlash uchun ko'plab imkoniyatlarni taqdim etadi. Bu, ma'lumotlarni shifrlash, ro'yxatga olish va boshqa xavfsizlik kutilmalarini o'z ichiga oladi.

Monitoring va dastur boshqarish: SQL Server, ma'lumotlar bazalarining monitoringini olib borish, xato va boshqa masalalarni aniqlash, avtomatik yuklanish va avtomatik tuzish imkonini ta'minlaydi. Shuningdek, SQL Server Management Studio (SSMS) kabi dastur boshqarish vositalari mavjud.

Tahlil va Business Intelligence (BI): SQL Server, ma'lumotlar tahlilini olib borish, reporting va BI ilovalarni yaratish imkonini beradi. Uning o'ziga xos manbalar va instrumentlar mavjud, shuningdek, Analysis Services, Reporting Services va Integration Services kabi muhim komponentlar taqdim etiladi.

Skalabilnost: SQL Server, kichik va o'rta hajmdagi tizimlardan, katta korporativ tizimlarga qadar bo'lgan skalabilnostga ega. Bu, ma'lumotlar bazasining tizimlarni kengaytirish va o'zgartirish imkonini ta'minlaydi.

SQL Server, qulay va kuchli ma'lumotlar bazasi tizimi bo'lib, Microsoft ekosistemasi bilan integratsiyasini osonlashtiradi. Bu esa dasturchilar va ma'lumotlar bazasi administratorlarining tajribalarini oshiradi va korporativ ma'lumotlarni samarali va xavfsiz boshqarishini ta'minlaydi.

Microsoft SQL Server va C# dasturlash tili o'rtasidagi integratsiya juda sodda va samarali bo'lib, ularga o'zaro hamkorlik qilish bilan dasturchilar tarmoq ilovalarini yaratishda yordam beradi. Quyidagi usullar orqali MS SQL Server va C# ni integratsiyalash mumkin:

ADO.NET: ADO.NET, C# dasturlash tilining standart ma'lumotlar manbasi bog'liqlik qo'llanmasi hisoblanadi. U SQL Server bilan bog'liq ma'lumotlarga murojat qilish va ularga o'zgartirishlar kiritishga imkon beradi. Bu, ma'lumotlarni o'qish, yozish, yangilash va o'chirish uchun samarali usul hisoblanadi.

Entity Framework (EF): Entity Framework, ob'ekt-ma'lumot modelini (ORM) ishlatadi va bog'liqlikni boshqa tillarga nisbatan yengillik qiladi. EF, C# dasturlash tilida ma'lumotlar bazasiga murojat qilish va ma'lumotlarni ishlatishda osonlik va samarali bo'lib, SQL Server bilan integratsiyasini osonlashtiradi.

LINQ (Language Integrated Query): LINQ, C# dasturlash tilining tajriba olishib borilayotgan qismi hisoblanadi. Bu, ma'lumotlar bazasidan ma'lumotlarni tanlash, qidirish va filtrlash uchun kuchli usulni taqdim etadi. LINQ, SQL Server bilan murojat qilish va ma'lumotlarni ishlatishda osonlik va osonlikni ta'minlaydi.

Stored Procedures va Functions: SQL Server daqlari (stored procedures) va funktsiyalari, ma'lumotlar bazasidagi xususiyatlarni C# dasturlash tili orqali ishlatishga imkon beradi. Bu, ma'lumotlarni ma'lumotlar bazasidan olish, o'zgartirishlar kiritish va saqlash jarayonlarini o'z ichiga oladi.

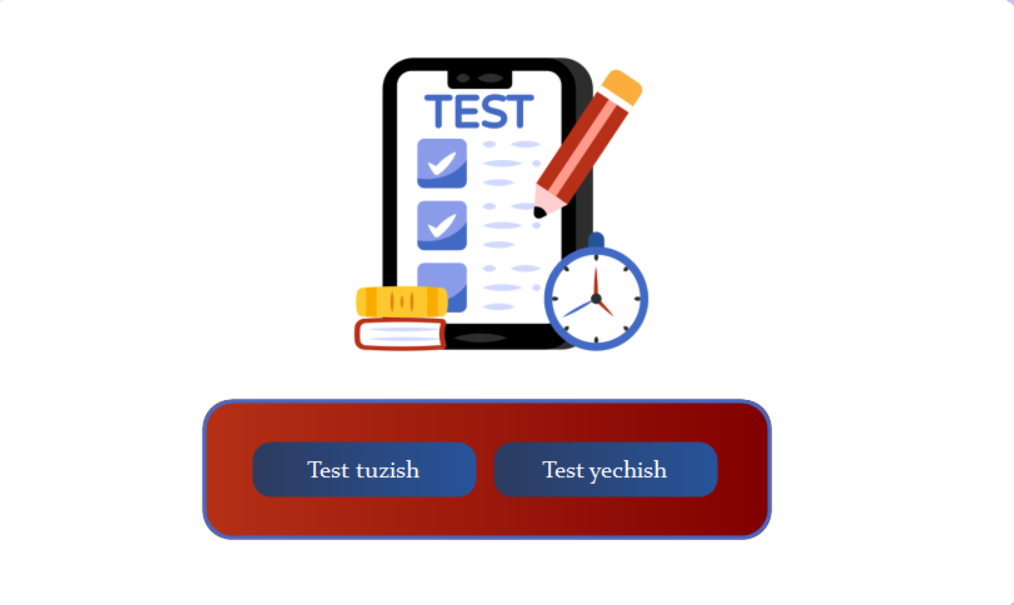
SQLCLR (SQL Server Common Language Runtime): SQLCLR, SQL Server ichidagi .NET dasturlashini ta'minlaydi. Bu, C# dasturlarini ma'lumotlar bazasida yaratish, boshqarish va ishlatishga imkon beradi. SQLCLR, ma'lumotlar bazasida qo'shimcha funktsiyalarni yaratish va ularga murojat qilish uchun juda qulaydir.

MS SQL Server va C# ni integratsiyalash orqali, dasturchilar korporativ tizimlarni ishlab chiqish va ularga bog'liqlikni oson va samarali qilishadi. Bu, ma'lumotlarni saqlash, o'qish, yozish, ma'lumotlarni tahlil qilish va boshqalar kabi kerakli vazifalarni bajarishda katta imkoniyatlar ta'minlaydi.

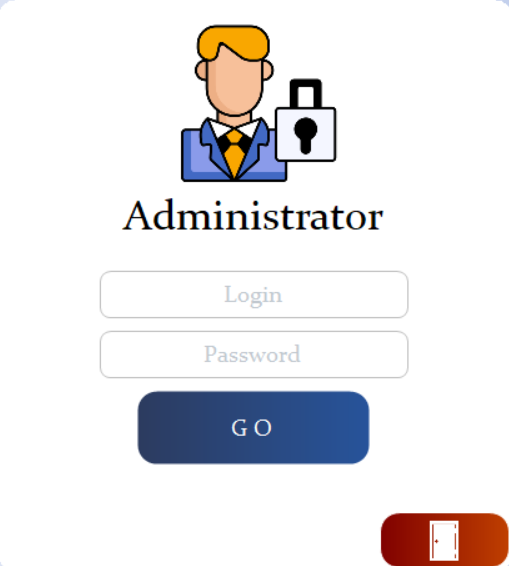
# II BOB. AVTOMATLASHTIRILGAN AXBOROT TIZIMINI YARATISH IMKONINI BERUVCHI DASTURIY TA’MINOT YARATISH

## 2.1. Loyihaning yaratilish qismi bilan tanishish

Ilovada asosan 9 ta qavat mavjud bo'lib, har qaysi o'ziga xos bo'lgan amallarni bajaradi. Bu qavatlar asosan 2 qismga bo`linadi.Admin va Talaba.Birinchi oynada quyidagicha korinishda:



Agar biz Test tuzish tugmasini tanlasak bizga admin qismi kelib chiqadi.



Bazaga kiritilgan Login Parol orqali admin kirsak, Adminning imkoniyatlarini ko`rib chiqamiz:

GunaUI ning TabControl komponentasi, Windows Forms ilovalarida sahifalarni yashirin qilish uchun ishlatiladi. Bu komponent, bir nechta tablar (yoki sahifalar) orqali aylanadi va foydalanuvchiga har bir tab orqali alohida ma'lumotlar qo'shish, o'qish yoki tahrirlash imkonini beradi. TabControl, dastur interfeysini yaxshi ko'rsatish uchun juda foydali bo'ladi, chunki foydalanuvchilar o'zlarining so'rashicha ma'lumotlarni qulayroq topishadi.

GunaUI ning TabControl komponentasi quyidagi imkoniyatlarga ega:

Alohida sahifalar:Har bir tab, alohida sahifani belgilaydi. Bu, foydalanuvchilarga bir nechta ma'lumotlar kiritish va ko'rish uchun ko'proq joy beradi.

Dinamik sahifalar: TabControl, dastur ishlayotgan paytda sahifalarni qo'shish, o'chirish va yangilash imkonini beradi. Bu, dinamik interfeys yaratish uchun juda foydali bo'ladi.

Stil va dizayn imkoniyatlari: TabControl, tablarning dizaynini o'zgartirish uchun keng imkoniyatlarga ega. Bu, foydalanuvchilarga moslashtirish va tajribalash imkonini beradi.

Ko'plab xususiyatlarga ega tablar: Har bir tab, o'z xususiyatlari va ko'rinishi bo'lishi mumkin. Bu, dasturni tuzatish va dizayn qilishda ko'plab variantlarga ega bo'lish uchun juda qulay.

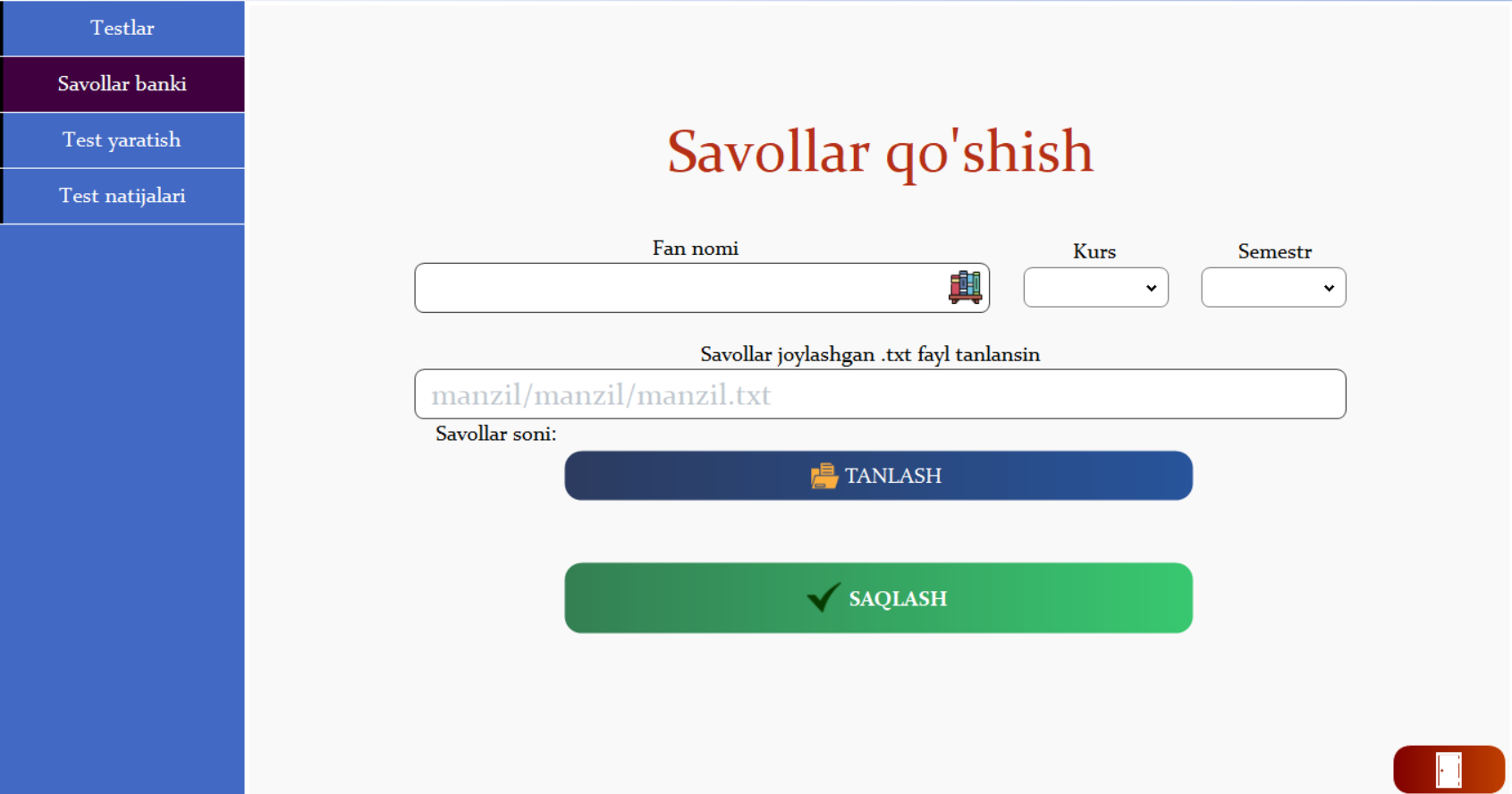
Tizim o'rnatish imkoniyatlari: TabControl, sahifalarni o'rnatish va joylashtirishga qulayliklar taqdim etadi. Bu, dasturni tuzatish va foydalanishni osonlashtirish uchun muhimdir.

GunaUI's TabControl, Windows Forms ilovalarida interfeys dizaynini yaxshilashda va foydalanuvchilar uchun yaxshi tajribani ta'minlashda qulay va foydali bo'ladi.

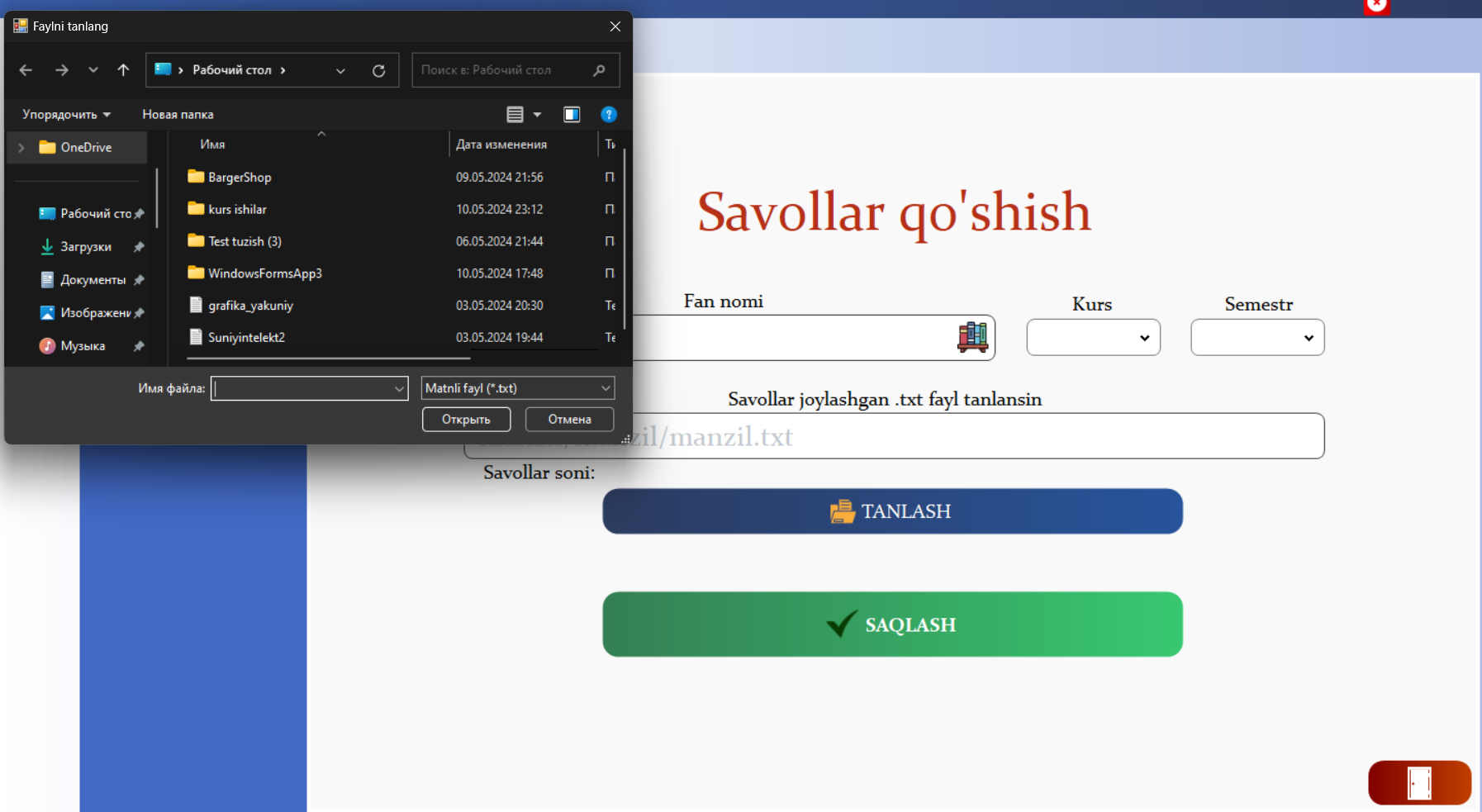
TabControlda quyidagilar joylashgan:

* Testlar;
* Savollar banki;
* Test yaratish;
* Test natijalari;

Har bir bandidagi oynalarda o`z vazifasiga ega bo`lgan kompanentalar mavjud.



Biz misol o`qituvchi va talaba misolida qaraydigan bo`lsak buyerda o`qituvchi fan bo`yicha yoki bo`lmasa ixtiroriy test savollarni tanlash orqali baza yaratadi ya’ni test savollaridan iborat bo`lgan baza keyinchalik,Ushbu tuzilma orqali foydalanuvchilar o'zlariga qulay va to'liq muvofiq interfeysda ishlashlari mumkin bo'ladi. Har bir qavat o'zining xizmat ko'rsatish uchun maxsus joylarga ega bo'lishlari mumkin.



C# da fayllar bilan ishlash. OpenFileDialog componentalari bilan ishlash

Fayllarni boshqarish ko'plab ma'lumotlarni qayta ishlash dasturlarining muhim jihati hisoblanadi. Zamonaviy texnologiyalar fayl tizimi bilan ishlashda foydalanuvchi qulayligini ta'minlash uchun turli xil vositalardan keng foydalanadi, jumladan OpenFileDialog

Fayllar bilan ishlashning asosiy tushunchalari:

1. Ma'lumotlar oqimlari

C# da fayllar bilan ishlashning asosiy mexanizmi ma'lumotlar oqimidir. Oqimlar ma'lumotlarni kiritish/chiqarish uchun mavhumlikni ta'minlaydi va katta hajmdagi ma'lumotlar bilan samarali ishlash imkonini beradi. Muayyan vazifalarga qarab qo'llanilishi mumkin bo'lgan kirish oqimlari va chiqish oqimlari kabi ko'plab oqim turlari mavjud.

2. Fayllar bilan ishlash sinflari

C# tilida fayllar va kataloglarni boshqarish uchun mo'ljallangan bir qancha sinflar mavjud. Masalan, File va Directory sinflari fayllar va kataloglarda o'qish, yozish, nusxa ko'chirish, ko'chirish va o'chirish operatsiyalarini bajarish uchun statik usullarni taqdim etadi.

3. Matnli va ikkilik fayllar bilan ishlash

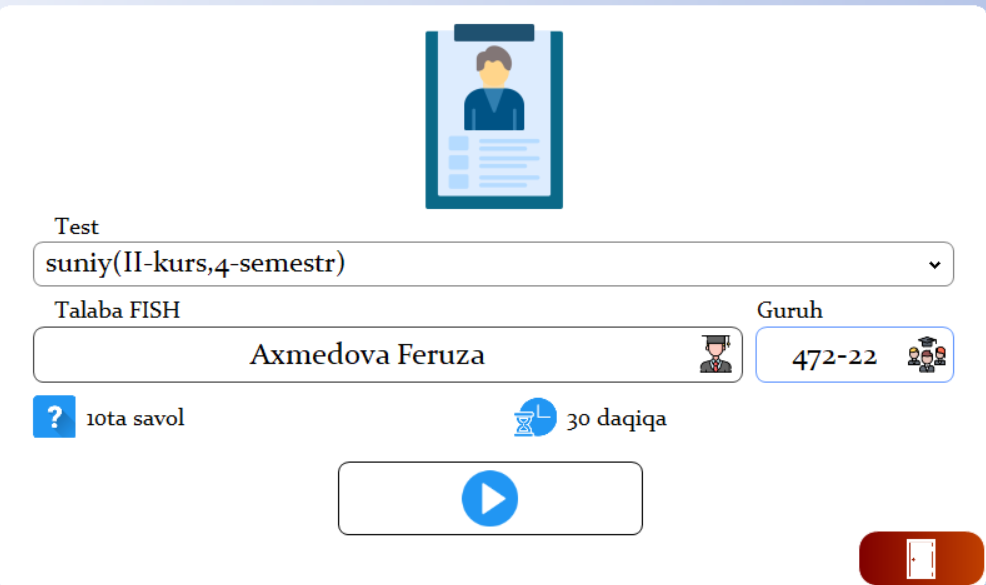
C# har xil turdagi fayllar, jumladan, matn va ikkilik fayllar bilan ishlash uchun vositalarni taqdim etadi. Matnli fayllarni o'qish va yozish uchun StreamReader va StreamWriter sinflaridan, ikkilik fayllar uchun esa BinaryReader va BinaryWriterdan foydalanishingiz mumkin. Bu sizga turli formatdagi ma'lumotlarni samarali qayta ishlash imkonini beradi.

4. Fayllar bilan ishlashda istisnolar bilan ishlash

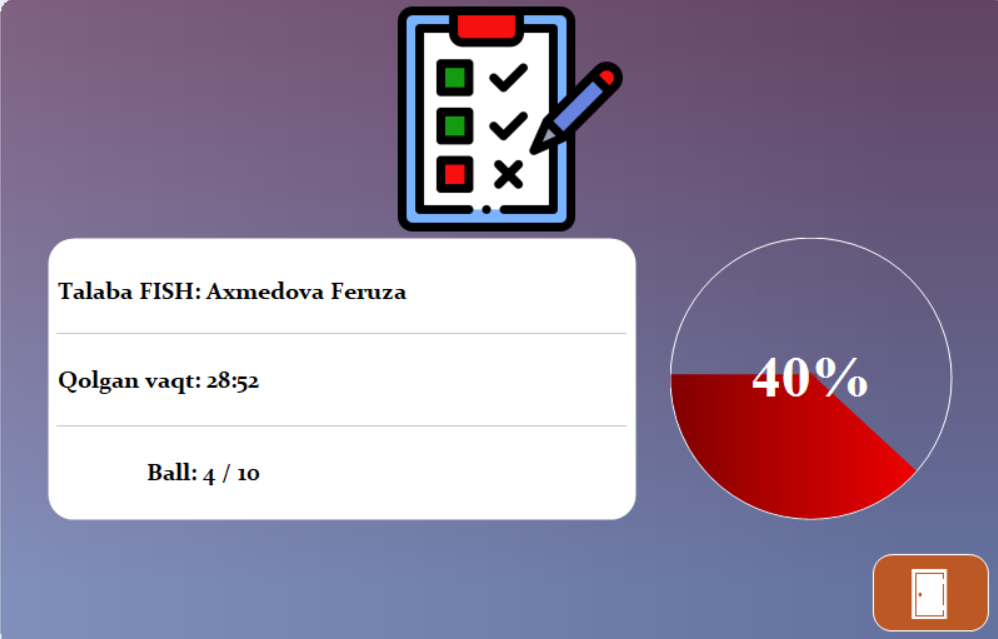
Fayllar bilan ishlashda, etishmayotgan fayl, kirish huquqlarining etarli emasligi va boshqalar kabi xatolar ehtimolini hisobga olish kerak. C# bunday vaziyatlar yuzaga kelganda dasturning ishdan chiqishini oldini olish uchun istisnolardan foydalanadi.



O`qituvchi kiritgan testlarini ko`rishimiz mumkin va biz Test yaratish qismida bundan ixtiroriy ravishda testlarni tuzishimiz mumkin ya’ni savollar soni nechta va nechi daqiqa vaqt berilishi haqida biz buyruq bersak bizning dastur shu bo`yicha ishlaydi hamda talabalarning o`zlashtirishini va bilimini sinash maqsadida test qilib ko`radigan bo`lsak bu qism ham faqat admin qismi yani o`qituvchiga ko`rinadi talabaga esa faqatgina ishlagan testini natijasi ko`rinadi xolos.



Quyida test ishlash uchun har bir talaba qaysi fannan test ishlayotganini tanlasa, bazadan shu fanning testi olinadi ya’ni u bu fan nechta savoldan iborat va qancha vaqt berilganligini ham shu yerda ko`rishimiz mumkin. Talaba o`z ism Familyasi va guruh nomeri orqali kiradi. Bular ham bazada saqlanib boradi va oxirida



Bunda test ishlagan talaba o`zining natijasini ko`rishi mumkin



Bular esa admin qisimida quyidagicha ko`rinadi

Bu tizimning bir qator qulayliklaridan biri hisoblanadi chunki,o`qituvchining talabalarni o`zlashtirishi muntazam kuzatib boorish ya’ni har bir ishlangan test sana vaqti va nechta ishlagani bilan saqlanib turadi.

Yana bu yerda bir qancha imkoniyatlar ya’ni Talaba ismi familyasi, guruh nomeri va bazaga kirilgan testlar nomi bilan qidiruv tizimi mavjud.Buning qulayligi shundaki ixtiyoriy vaqtda o`qituvchi talabalarning o`zlashtirishini kuzatib nazorat qilib borishi mumkin.

## 2.2. Yaratilgan tizimining amaliy ahamiyati va qo’llanish sohalari

Windows Forms muhida avtomatlashtirilgan axborot tizimi, ta'lim muassasalarida yoki kurslarda sinov va imtihon tuzish jarayonlarini avtomatlashtirish uchun o'zgaruvchiliklar yaratish imkonini beradi. Bu tizim orqali biletlar ustida ishlashga yo'lliklar, fanlar bo'yicha test yaratishga yordam beruvchi usullar va imtihon natijalarini o'qib chiqish va analiz qilish jarayonlarini takomillashtirish imkoniyatiga ega.

Ta'lim muassasalari va universitetlar: o'quv jarayonlarini boshqarish va o'quvchilar bilim darajasini baholashni avtomatlashtirish uchun bizning loyihamiz foydali bo'ladi.

Korporativ ta'lim tizimlari: korporativ kompaniyalarda xodimlar uchun o'quv kurslari tashkill etish va ularni imtihon asosida baholashni avtomatlashtirish uchun foydalanish mumkin.

Darsliklar va o'quv materiyallar muassasalari: darsliklar, yangi o'quv materiallar va dasturlar yaratish va ularga imtihon testlarini qo'llash imkoniyatini takomillashtirish uchun foydalanish mumkin.

Avtomatlashtirilgan axborot tizimi orqali fan bo`yicha testlarni yaratish va boshqarish, o'quvchilarning bilim darajasini baholash va ta'lim jarayonlarini takomillashtirishga yordam beradi. Bu tizimlar ta'lim sohasida texnologik o'sish va ta'limga zarur muammo va vazifalarni hal qilishda muhim ahamiyatga ega.

Oliy ta’lim o‘quv jarayonini tashkil etishda innovatsion texnologiyalarning roli kun sayin ortib bormoqda. Masofaviy texnologiyalardan foydalanish zamonaviy ta’limning imkoniyatlarini yanada kengaytirdi. Bugungi kunda Yer kurrasining istalgan joyidan turib, zamonaviy axborot-kommunikatsiya texnologiyalari (AKT) imkoniyatidan foydalangan holda ta’lim olish mumkin. Zero an’anaviy ta’lim o‘z mavqeini saqlab tursa ham, keyingi paytlarda masofaviy o‘qitish texnologiyalari kundan-kun ommaviylashib bormoqda.

Bugungi kunda mamlakatimizda yangi jahon axborot-ta’lim muhitiga integrallashishga yo‘naltirilgan ta’lim tizimi barpo etilmoqda. Bu ta’lim jarayonini tashkil etishda zamonaviy texnik imkoniyatlarga javob beradigan sezilarli o‘zgarishlar bilan kuzatilmoqda. Zamonaviy axborot texnologiyalarining ta’lim sohasiga kirib kelishi ta’lim usullari va o‘qitish jarayonini yangicha yondashuv asosida tashkil etish shakllarini sifatli ravishda qulaylashtirib, o‘zgartirish imkonini bermoqda. Axborot-kommunikatsiya texnologiyalari ta’lim tizimini modernizatsiyalashtirish jarayonining eng muhim qismidir. AKT — bu turli texnik va dasturiy qurilmalar bilan axborotga ishlov berish usullaridir. U birinchi navbatda, zarur dasturiy ta’minotga ega bo‘lgan kompyuterlar va ma’lumotlar joylashtirilgan telekommunikatsiya vositalaridir.

Elektron va masofaviy texnologiyalar — ta’limning axborot va kommunikatsiya texnologiyalari qo‘llangan variantlaridir.

Elektron ta’lim (E-Learning) — avval «Elektron ta’lim» atamasi kompyuter yordamida o‘qitish deb tushunilgan, biroq axborot texnologiyalari rivoji bilan bu tushuncha yanada kengaytirildi. Bugungi kunda elektron ta’lim ko‘pgina ta’lim texnologiyalarini qamrab olmoqda, ularni shartli ravishda, 2 xil turga, ya’ni sinxron va asinxron turlarga bo‘lish mumkin.

Sinxron elektron ta’lim — masofaviy ta’lim hisoblanadi, lekin bu real vaqtda amalga oshiriladigan ta’limdir. U oddiy kunduzgi ta’limga o‘xshaydi, farqi shundaki, ishtirokchilar bir-biridan uzoq masofada bo‘ladi. Kundan-kunga keng tarqalib borayotgan vebinarlar mazkur ta’lim shaklining eng yorqin ko‘rinishidir. Ma’ruzalarni tashkillashtirishda maxsus dasturiy ta’minotlar qo‘llaniladi.

Asinxron elektron ta’lim — bu talaba barcha kerakli ma’lumotni onlayn-manbalardan yoki elektron axborot tashish vositalari (CD, DVD yoki flash-kartalar)dan olishi va materialni o‘zlashtirish sur’ati va jadvalini o‘zi mustaqil tashkil etishdir. Asinxron elektron ta’lim tizimiga barcha turdagi CD-kurslar va elektron o‘qitish kurslari, ostkastlar vaskrinkastlar kiradi. Bugungi kunda elektron ta’lim ko‘pchilik OTM’larda ta’lim jarayonining ajralmas qismi bo‘lib qolgan, u shuningdek, malaka oshirish kurslarini tashkil etishda ham o‘z o‘rnini topgan, ba’zi korporatsiyalarda bo‘linmalar mavjud bo‘lib, ularning vazifasi xizmatchilar uchun elektron kurslar tashkil etishdir.

Masofaviy ta’lim texnologiyalari — masofaviy ta’lim bu E-Learningga qaraganda kengroq tushunchadir, u interfaol mustaqil ta’limning va qo‘llab-quvvatlashning intensiv maslahat sintezi hisoblanadi. Shunday qilib, elektron ta’lim masofaviy ta’limning bir bo‘lagi hisoblanadi. Masofaviy ta’lim asosiy o‘quv materialini o‘quvchilarga yetkazib berish va o‘quv jarayonida o‘quvchi va o‘qituvchi orasida interfaol ishlashni ta’minlaydi. Bunda qo‘llanmalarni yetkazib berish kompyuter va Internetsiz ham amalga oshirilishi mumkin.

Masofaviy texnologiyalardan foydalangan holda o‘qitishning juda ko‘p ijobiy tomonlari mavjud.

O‘qish va ishni birga qo‘shib olib borish — o‘quvchilar ishdan ajralmagan holda ta’lim olish imkoniga ega bo‘ladilar, bu, ayniqsa, malaka oshirish yoki ikkinchi oliy ma’lumot oluvchilarga juda qo‘l keladi.

Sifatli texnologiyalar va o‘quv mazmunini egallash — talaba sifatli o‘quv materiallari yordamida o‘qitilishi, o‘qituvchi bilan muloqat qilishi va o‘z individual o‘quv rejasini tuzishi mumkin.

Baholashning xolisligi — masofaviy ta’lim texnologiyalari bilim sifatining doimiy nazorati, natijalarning baholanishi, inson omilidan xoli bo‘lgan xolis avtomatlashtirilgan baholash joylarda moddiy manfaatdorlikni yo‘qotishni ko‘zda tutadi.

Ta’limda individual yondashuv — o‘zgaruvchan grafik, ish va o‘qishni birga qo‘shib olib borish, shuningdek, o‘zlashtirilayotgan materialni ma’lumotni individual o‘zlashtirish tezligiga moslash masofaviy ta’limni barcha uchun qulay qilib qo‘ymoqda.

Bizning yaratgan ilovamiz ham hozirgi kun talabiga aylanayotgan texnologiyalarning bir qismini tashkil qiladi. Bunda yoshlarning ya’ni hzoirgi kunda Ta’lim tizimidagi o’zgarishlardan ko`radigan bo`lsak, masofaviy ta’lim turi ham jadal suratlar bilan rivojlanib bormoqda. Yoshlarga yaratilayotgan imkoniyatlar, xususan manashu sohadagi izlanishlarni misol qiladigan bo`lsak, turli xil elektron darsliklar hamda o`quv qo`llanmalar foydalanish uchun taqdim qilngan. Masofaviy ta’lim tizimida o`qiyotgan talabalarni bilimini sinash maqsadida bizning ilovamiz qo`l keladi va o`qituvchilarga talabalarning o`zlashtirishini nazorat qilishda yordam beradi.

# XULOSA

Xulosa qilib aytganda ushbu loyiha avtomatlashtirilgan tizim yaratish uchun katta potensialga ega. Ushbu tizim o‘quv jarayonini, kadrlar bo‘limini, test tashkil etishni va dasturlar tanlashni samarali boshqarishga yordam beradi. Buning yanada ko‘proq foydali taraflari o‘quvchilar va o‘qituvchilar uchun imtihon tayyorlashda va tashkilotlarga kadrlar bilan imtihonlar o‘tkazishda o‘zini ko‘rsatishi mumkin. Ushbu loyihadan foydalanish orqali, tashkilotlar o‘zlarining kadrlarini tahlil qilish, yangi malakali kadrlarni qabul qilish va xodimlarni talablariga mos ravishda rivojlantirishda qulayliklar yaratishlari mumkin. Bu loyiha texnologiya sohasida ishlab chiqilgan bo‘lib, dasturlash, ma’lumotlar bazalari, va texnologik muammolar orasidagi avtomatlashtirilgan jarayonlarni o‘rganuvchilar uchun qulayliklar yaratadi. Tizim foydalanuvchilarga ularning tanlovlariga va talablariiga mos ravishda imtihonlar tayyorlash imkonini beradi. Bu, har bir foydalanuvchi uchun maxsus imtihonlarni generatsiya qilish va natijalarni baholash imkoniyatini ta’minlaydi va natijalarni kuzatib borishga imkoniyat yaratadi hamda o‘quv jarayonini, test tashkil etishni va natijalarni baholashni avtomatiklashtiradi. Bu, kadrlar bo‘limlarining vaqt va energiyasini tejamkorlik bilan ishlatishga imkon beradi, ular testlar tayyorlash va natijalarni baholashga ajratilgan vaqtni boshqa faoliyatlarga ajratishlari mumkin.O‘quvchilar va xodimlar individual ravishda testlar o‘tkazib, o‘zlarining bilimlarini sinovdan o‘tkazishadi va shu orqali rivojlantirish kerak bo‘lgan yo‘nalishlarini aniqlaydi.Mana o`quv yili oxirlab sessiyalarga ham navbat keldi, bu esa talabalarni bor bilimini sinash vaqti digani. Shunday vaqtda bu dastur bizga yordam beradi chunki, har bir testlar to`plamini biz bazamizga yuklab qo`ysak unday xoxlagan payt kirib ishlagan holda o`z bilimimizni sinab ko`rishimiz mumkin. Bu orqali biz o`rganilgan bilimni sinovdan o`tkazamiz va mustahkamlab boramiz.

# FOYDALANILGAN ADABIYOTLAR

1. Герберт Шилдт. C# 4.0 Полноеруководство. Москва. Санкт-Петербург. Киев. 2011, 473-б.
2. Джозеф Албахари и Бен Албахари. C# 6.0 Справочник. Полное описание языка.Москва. Санкт-Петербург. Киев. 2016, 151-б.
3. Борис Пахомов. C# для начинающих. Санкт-Петербург«БХВ-Петербург». 2014, 293-б.
4. А.Л.Марченко. C#. Введение в программирование. Издательство Московского университета. 2005, 88-б.
5. Ilmiy-texnologik kitoblar. Muallif: Sells C.
6. Основы систем баз данных: Учебник. / Е. Ж. Айтхожаева – Алматы: КазНИТУ имени К. И. Сатпаева, 2016. – 279 с. Ил. 38. Табл. 8. Библиогр. –53 назв.
7. Бейли Л. Изучаем SQL /Пер. с англ. – СПб.: Питер, 2012. – 573 с.
8. 4.Object orentiring programming in C#. 2016
9. Пахомов Б. И. C# для начинающих. — СПб.БХВ-Петербург, 2014. — 432 с.
10. Справочник. Полное описание языка, 6-е изд.:Пер. с англ. – М.: ООО ―И. Д. Вильямс‖, 2016. –
11. 1040.
12. Герберт Шилдт. C# 4.0: полное руководство.:Пер. с англ. М.: ООО ―И.Д. Вильямс‖,
13. 2011. 1056 с.

# ILOVALAR

namespace Test\_tuzish

{

class Query

{

static string path = Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10);

SqlConnection sqlConnection = new SqlConnection($@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename={path}\Data.mdf;Integrated Security=True");

public DataTable Select(string query)

{

sqlConnection.Open();

SqlCommand selectcmd = new SqlCommand(query, sqlConnection);

DataTable dataTable = new DataTable();

dataTable.Load(selectcmd.ExecuteReader());

sqlConnection.Close();

return dataTable;

}

public void Insert(string query)

{

try

{

sqlConnection.Open();

SqlCommand insertcmd = new SqlCommand(query, sqlConnection);

insertcmd.ExecuteNonQuery();

sqlConnection.Close();

}

catch (Exception ex)

{

throw ex;

}

}

public void Update(string query)

{

sqlConnection.Open();

SqlCommand updatecmd = new SqlCommand(query, sqlConnection);

updatecmd.ExecuteNonQuery();

sqlConnection.Close();

}

public void Delete(string query)

{

sqlConnection.Open();

SqlCommand deletecmd = new SqlCommand(query, sqlConnection);

deletecmd.ExecuteNonQuery();

sqlConnection.Close();

}

public void FillComboBox(string col, string table, ComboBox comboBox)

{

comboBox.Items.Clear();

var data = Select($"select {col} from {table}");

for (int i = 0; i < data.Rows.Count; i++)

{

comboBox.Items.Add(data.Rows[i][0]);

}

}

public void FillListBox(string col, string table, ListBox listbox)

{

listbox.Items.Clear();

var data = Select($"select {col} from {table}");

for (int i = 0; i < data.Rows.Count; i++)

{

listbox.Items.Add(data.Rows[i][0]);

}

}

}

}

**TxtReader class**

namespace Test\_tuzish

{

internal class TxtReader

{

public static void CopyFile(string oldPath, string newPath)

{

try

{

File.Copy(oldPath, newPath);

File.SetAttributes(newPath, File.GetAttributes(newPath) | FileAttributes.ReadOnly);

}

catch

{

MessageBox.Show("Fayl topilmadi", "Xato!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

public static Dictionary<string, (int correctIndex, List<string> answers)> ReadQuestions(string filePath, Guna2HtmlLabel infoLabel)

{

Dictionary<string, (int correctIndex, List<string> answers)> questionsDict = new Dictionary<string, (int correctIndex, List<string> answers)>();

string currentQuestion = "";

List<string> currentAnswers = new List<string>();

int correctIndex = -1;

if (!File.Exists(filePath))

{

infoLabel.Text = "Fayl mavjud emas";

infoLabel.ForeColor = Color.Red;

return null;

}

string[] lines = File.ReadAllLines(filePath);

foreach (string line in lines)

{

if (line.StartsWith("#"))

{

if (!string.IsNullOrEmpty(currentQuestion) && currentAnswers.Count > 0)

{

if (!questionsDict.ContainsKey(currentQuestion))

{

questionsDict.Add(currentQuestion, (correctIndex, currentAnswers));

currentAnswers = new List<string>();

correctIndex = -1;

}

}

// Устанавливаем новый текущий вопрос

currentQuestion = line.TrimStart('#');

}

else if (line.StartsWith("+")) // Если строка начинается с "+", это правильный ответ

{

currentAnswers.Add(line.TrimStart('+'));

correctIndex = currentAnswers.Count - 1;

}

else if (line.StartsWith("-")) // Если строка начинается с "-", это неправильный ответ

{

currentAnswers.Add(line.TrimStart('-'));

}

else if (string.IsNullOrWhiteSpace(line))

{

continue;

}

else

{

infoLabel.Text = "Savollar notog'ri tuzilgan";

infoLabel.ForeColor = Color.Red;

return null;

}

}

if (correctIndex == -1)

{

infoLabel.Text = "Savollar orasida tog'ri javob boshida '+' belgisi bo'lsin";

infoLabel.ForeColor = Color.Red;

return null;

}

if (!string.IsNullOrEmpty(currentQuestion) && currentAnswers.Count > 0)

{

questionsDict.Add(currentQuestion, (correctIndex, currentAnswers));

}

return questionsDict;

}

public static void WriteToJSONSubjectAndNewPath(string subjectName, string newfilePath)

{

string JSONPath = $"{Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10, 10)}/data.json";

string jsonFromFile = File.ReadAllText(JSONPath);

var existingData = JsonSerializer.Deserialize<Dictionary<string, string>>(jsonFromFile);

existingData.Add(subjectName, newfilePath);

var options = new JsonSerializerOptions

{

Encoder = System.Text.Encodings.Web.JavaScriptEncoder.UnsafeRelaxedJsonEscaping,

WriteIndented = true

};

string jsonString = JsonSerializer.Serialize(existingData, options);

File.WriteAllText(JSONPath, jsonString);

MessageBox.Show("Saqlandi", "Muvaffaqiyatli", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

public static void FillSubjectsCombobox(Guna2ComboBox comboBox)

{

comboBox.Items.Clear();

string JSONPath = $"{Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10, 10)}/data.json";

if (File.Exists(JSONPath))

{

string jsonFromFile = File.ReadAllText(JSONPath);

var dictionary = JsonSerializer.Deserialize<Dictionary<string, string>>(jsonFromFile);

foreach (var key in dictionary.Keys)

{

comboBox.Items.Add(key);

}

}

else

{

MessageBox.Show("data.json fayli topilmadi", "Xatolik", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

public static void FillRichTextBox(RichTextBox richTextBox, string key)

{

string JSONPath = $"{Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10, 10)}/data.json";

if (File.Exists(JSONPath))

{

string jsonFromFile = File.ReadAllText(JSONPath);

var dictionary = JsonSerializer.Deserialize<Dictionary<string, string>>(jsonFromFile);

string allText = File.ReadAllText(dictionary[key]);

richTextBox.Text = allText;

}

else

{

MessageBox.Show("data.json fayli topilmadi", "Xatolik", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

public static void RemoveJSONElementWithKey(string key)

{

string filePath = $"{Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10, 10)}/data.json";

if (File.Exists(filePath))

{

string jsonFromFile = File.ReadAllText(filePath);

var dictionary = JsonSerializer.Deserialize<Dictionary<string, string>>(jsonFromFile);

dictionary.Remove(key);

var options = new JsonSerializerOptions

{

Encoder = System.Text.Encodings.Web.JavaScriptEncoder.UnsafeRelaxedJsonEscaping,

WriteIndented = true

};

string jsonString = JsonSerializer.Serialize(dictionary, options);

File.WriteAllText(filePath, jsonString);

}

else

{

MessageBox.Show("data.json fayli topilmadi", "Xatolik", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

public static Dictionary<string, (string TrueAnswer, List<string> answers)> GetQuestionsDic(string filePath)

{

Dictionary<string, (string TrueAnswer, List<string> answers)> questionsDict = new Dictionary<string, (string TrueAnswer, List<string> answers)>();

string currentQuestion = "";

List<string> currentAnswers = new List<string>();

string TrueAnswer = "";

if (!File.Exists(filePath))

{

return null;

}

string[] lines = File.ReadAllLines(filePath);

foreach (string line in lines)

{

if (line.StartsWith("#"))

{

if (!string.IsNullOrEmpty(currentQuestion) && currentAnswers.Count > 0)

{

if (!questionsDict.ContainsKey(currentQuestion))

{

questionsDict.Add(currentQuestion, (TrueAnswer, currentAnswers));

currentAnswers = new List<string>();

TrueAnswer = "";

}

}

// Устанавливаем новый текущий вопрос

currentQuestion = line.TrimStart('#');

}

else if (line.StartsWith("+")) // Если строка начинается с "+", это правильный ответ

{

currentAnswers.Add(line.TrimStart('+'));

TrueAnswer = line.TrimStart('+');

}

else if (line.StartsWith("-")) // Если строка начинается с "-", это неправильный ответ

{

currentAnswers.Add(line.TrimStart('-'));

}

else if (string.IsNullOrWhiteSpace(line))

{

continue;

}

else

{

return null;

}

}

if (TrueAnswer == "")

{

return null;

}

if (!string.IsNullOrEmpty(currentQuestion) && currentAnswers.Count > 0)

{

questionsDict.Add(currentQuestion, (TrueAnswer, currentAnswers));

}

return questionsDict;

}

public static string GetPathOfTest(string key)

{

string JSONPath = $"{Environment.CurrentDirectory.Remove(Environment.CurrentDirectory.Length - 10, 10)}/data.json";

if (File.Exists(JSONPath))

{

string jsonFromFile = File.ReadAllText(JSONPath);

var dictionary = JsonSerializer.Deserialize<Dictionary<string, string>>(jsonFromFile);

return dictionary[key];

}

else

{

MessageBox.Show("data.json fayli topilmadi", "Xatolik", MessageBoxButtons.OK, MessageBoxIcon.Error);

return null;

}

}

public static void DeleteTxtFile(string filePath)

{

try

{

FileAttributes attributes = File.GetAttributes(filePath);

if ((attributes & FileAttributes.ReadOnly) == FileAttributes.ReadOnly)

{

attributes &= ~FileAttributes.ReadOnly;

File.SetAttributes(filePath, attributes);

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при снятии атрибута 'только для чтения': " + ex.Message, "Ошибка");

return;

}

try

{

File.Delete(filePath);

MessageBox.Show("O'chirildi.", "Muvaffaqiyat");

}

catch {}

}

}

}

**AdminPanel**

namespace Test\_tuzish

{

public partial class AdminPanel : UserControl

{

private MainForm mainForm;

private Query q = new Query();

private Dictionary<string, (int correctIndex, List<string> answers)> questions = null;

private int selected\_id = -1;

private string selected\_subject = "";

private int selected\_count = 0;

private int selected\_time = 0;

public AdminPanel(MainForm mainForm)

{

this.mainForm = mainForm;

InitializeComponent();

TxtReader.FillSubjectsCombobox(subjectCombobox);

RefreshDataGridViewT();

FillResult();

q.FillComboBox("DISTINCT studentGroup", "Results", studentGroupBox);

q.FillComboBox("DISTINCT testname", "Results", subjectNameBoxR);

studentGroupBox.Items.Insert(0, "");

subjectNameBoxR.Items.Insert(0, "");

}

private string file\_name = "";

private string file\_path = "";

private void guna2GradientButton1\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

file\_path = openFileDialog1.FileName;

filePathTxtBox.Text = file\_path;

file\_name = openFileDialog1.SafeFileName.Replace(".txt","");

questions = TxtReader.ReadQuestions(file\_path, infoLabel);

if (questions != null)

{

infoLabel.Text = "Savollar soni: " + questions.Count;

infoLabel.ForeColor = Color.Black;

}

}

}

private void guna2GradientButton2\_Click(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(subjectName.Text))

{

subjectName.Focus();

errorLabel.Visible = true;

}

else if (string.IsNullOrWhiteSpace(kursBox.Text))

{

kursBox.Focus();

errorLabel.Visible = true;

}

else if (string.IsNullOrWhiteSpace(semestrBox.Text))

{

semestrBox.Focus();

errorLabel.Visible = true;

}

else if (string.IsNullOrWhiteSpace(filePathTxtBox.Text))

{

filePathTxtBox.Focus();

errorLabel.Visible = true;

}

else if (infoLabel.ForeColor == Color.Red)

{

if(MessageBox.Show("Savollar joylashgan .txt fayl tanlansin", "Diqqat!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

guna2GradientButton1\_Click(sender, e);

}

}

else

{

string newfilePath = $"{Environment.CurrentDirectory}\\Savollar\\{file\_name + DateTime.Now.ToString(" dd-MM-yyyy HH-mm-ss") + ".txt"}";

TxtReader.CopyFile(file\_path, newfilePath);

TxtReader.WriteToJSONSubjectAndNewPath(subjectName.Text+"("+kursBox.Text+"-kurs,"+semestrBox.Text+"-semestr)", newfilePath);

TxtReader.FillSubjectsCombobox(subjectCombobox);

subjectName.Text = filePathTxtBox.Text = "";

kursBox.SelectedIndex = -1;

semestrBox.SelectedIndex = -1;

errorLabel.Visible = false;

}

}

private void subjectCombobox\_SelectedIndexChanged(object sender, EventArgs e)

{

if (subjectCombobox.SelectedIndex != -1)

{

TxtReader.FillRichTextBox(filesTextRichTextBox,subjectCombobox.SelectedItem.ToString());

}

}

private void guna2GradientButton2\_Click\_1(object sender, EventArgs e)

{

var table = q.Select($"SELECT COUNT FROM TESTS WHERE SUBJECT = '{subjectCombobox.SelectedItem}'");

if (subjectCombobox.SelectedItem != null)

{

var Qcount = TxtReader.GetQuestionsDic(TxtReader.GetPathOfTest(subjectCombobox.SelectedItem.ToString())).Keys.Count;

if (table.Rows.Count > 0)

{

countErrorLabel.Visible = true;

countErrorLabel.Text = $"{subjectCombobox.SelectedItem} avval qo'shilgan";

}

else if (Qcount < questionsCountNumeric.Value)

{

countErrorLabel.Visible = true;

countErrorLabel.Text = "Ushbu savollar bankida savollar yetarli emas";

}

else

{

countErrorLabel.Visible = false;

q.Insert($"INSERT INTO TESTS VALUES('{subjectCombobox.SelectedItem.ToString().Replace("'", "`")}',{questionsCountNumeric.Value},{testTimeNumeric.Value})");

MessageBox.Show("Muvaffaqiyatli", "Muvaffaqiyatli", MessageBoxButtons.OK, MessageBoxIcon.Information);

filesTextRichTextBox.Clear();

subjectCombobox.SelectedIndex = -1;

questionsCountNumeric.Value = 20;

testTimeNumeric.Value = 30;

}

}

else

{

countErrorLabel.Visible = true;

countErrorLabel.Text = "Savollar banki tanlansin";

}

}

private void RefreshDataGridViewT()

{

while (dataTableDGV.Rows.Count > 0)

{

dataTableDGV.Rows.RemoveAt(0);

}

dataTableDGV.DataSource = q.Select("SELECT id,subject as \"Fan nomi(Test nomi)\",count as \"Biletdagi savollar soni\",time as \"Ajratilgan vaqt(daqiqa)\" FROM TESTS");

dataTableDGV.Columns[0].Width = 80;

dataTableDGV.Columns[1].Width = 550;

dataTableDGV.ColumnHeadersHeight = 100;

}

private void dataTableDGV\_CellClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0 && e.ColumnIndex >= 0)

{

selected\_id = int.Parse(dataTableDGV.Rows[e.RowIndex].Cells[0].Value.ToString());

selected\_subject = dataTableDGV.Rows[e.RowIndex].Cells[1].Value.ToString();

selected\_count = int.Parse(dataTableDGV.Rows[e.RowIndex].Cells[2].Value.ToString());

selected\_time = int.Parse(dataTableDGV.Rows[e.RowIndex].Cells[3].Value.ToString());

}

}

private void guna2TabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (guna2TabControl1.SelectedIndex == 0)

{

RefreshDataGridViewT();

}

else if (guna2TabControl1.SelectedIndex == 3)

{

q.FillComboBox("DISTINCT studentGroup", "Results", studentGroupBox);

q.FillComboBox("DISTINCT testname", "Results", subjectNameBoxR);

studentGroupBox.Items.Insert(0, "");

subjectNameBoxR.Items.Insert(0, "");

}

foreach (TabPage tabPage in guna2TabControl1.TabPages)

{

if (guna2TabControl1.SelectedTab != tabPage)

{

ClearAllControls(tabPage.Controls);

}

}

}

private void ClearAllControls(ControlCollection controls)

{

foreach (Control c in controls)

{

if (c is TextBox)

{

((TextBox)c).Clear();

}

else if (c is RichTextBox)

{

((RichTextBox)c).Clear();

}

else if (c is ComboBox)

{

((ComboBox)c).SelectedIndex = -1;

}

else if (c is NumericUpDown)

{

((NumericUpDown)c).Value = 20;

}

}

}

private void guna2GradientButton2\_Click\_2(object sender, EventArgs e)

{

guna2TabControl1.SelectedTab = createTestPage;

}

private void guna2GradientButton3\_Click(object sender, EventArgs e)

{

if (selected\_id != -1)

{

if (MessageBox.Show($"«{selected\_subject} | {selected\_count}savol | {selected\_time}daqiqa» testini o'chirishga aminmisiz?", "Diqqat", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

q.Delete($"DELETE FROM TESTS WHERE ID = {selected\_id}");

MessageBox.Show("Muvaffaqiyatli", "Muvaffaqiyatli", MessageBoxButtons.OK, MessageBoxIcon.Information);

RefreshDataGridViewT();

}

}

}

private void exitButtons\_Click(object sender, EventArgs e)

{

mainForm.Menu\_Open();

}

private void FillResult()

{

string query = $"SELECT STUDENTNAME AS 'Talaba', STUDENTGROUP AS 'Guruh'," +

" TESTNAME AS 'Fan nomi(Test nomi)', " +

"BALL AS 'Ball',DATE\_TIME AS 'Test topshirilgan sana' " +

$"FROM RESULTS WHERE STUDENTNAME LIKE '%{studentName.Text}%'";

if (!string.IsNullOrWhiteSpace(studentGroupBox.Text))

{

query += $" AND STUDENTGROUP = '{studentGroupBox.Text}'";

}

if (!string.IsNullOrWhiteSpace(subjectNameBoxR.Text))

{

query += $" AND TESTNAME = '{subjectNameBoxR.Text}'";

}

resultDGV.DataSource = q.Select(query);

resultDGV.Columns[0].Width = 275;

resultDGV.Columns[1].Width = 110;

resultDGV.Columns[2].Width = 315;

resultDGV.Columns[3].Width = 100;

resultDGV.ColumnHeadersHeight = 100;

}

private void studentName\_TextChanged(object sender, EventArgs e)

{

FillResult();

}

private void studentGroupBox\_SelectedIndexChanged(object sender, EventArgs e)

{

FillResult();

}

private void subjectNameBoxR\_SelectedIndexChanged(object sender, EventArgs e)

{

FillResult();

}

private void deleteTestSourceBtn\_Click(object sender, EventArgs e)

{

if (subjectCombobox.SelectedIndex != -1)

{

if (MessageBox.Show($"«{subjectCombobox.SelectedItem}» savollar bankini o'chirish natijasida unga tegishli yaratilgan test ham o'chib ketadi, aminmisiz?", "Diqqat", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

string path = TxtReader.GetPathOfTest(subjectCombobox.SelectedItem.ToString());

TxtReader.DeleteTxtFile(path);

TxtReader.RemoveJSONElementWithKey(subjectCombobox.SelectedItem.ToString());

q.Delete($"DELETE FROM TESTS WHERE SUBJECT = '{subjectCombobox.SelectedItem}'");

TxtReader.FillSubjectsCombobox(subjectCombobox);

filesTextRichTextBox.Clear();

}

}

}

private void dataTableDGV\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

}

}

**Login**

namespace Test\_tuzish

{

public partial class LogIn : UserControl

{

private MainForm mainForm;

Query q = new Query();

public LogIn()

{

InitializeComponent();

}

public LogIn(MainForm mainForm)

{

this.mainForm = mainForm;

InitializeComponent();

}

private void guna2GradientButton1\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrWhiteSpace(loginTxt.Text) || !string.IsNullOrWhiteSpace(parolTxt.Text))

{

if (int.Parse(q.Select($"SELECT COUNT(\*) FROM ADMINS WHERE LOGIN = '{loginTxt.Text}' AND PAROL = '{parolTxt.Text}'").Rows[0][0].ToString()) > 0)

{

mainForm.AdminPanel\_Open();

}

else

{

ErrorLabel.Text = "Login yoki parol notog'ri";

ErrorLabel.Visible = true;

}

}

else

{

ErrorLabel.Text = "Maydonlarni to'ldiring";

ErrorLabel.Visible = true;

}

}

private void exitBtn3\_Click(object sender, EventArgs e)

{

mainForm.Menu\_Open();

}

}

}

**Menu**

namespace Test\_tuzish

{

public partial class Menu : UserControl

{

MainForm mainForm = null;

public Menu()

{

InitializeComponent();

}

public Menu(MainForm mainForm)

{

this.mainForm = mainForm;

InitializeComponent();

}

private void guna2GradientButton1\_Click(object sender, EventArgs e)

{

mainForm.LogIn\_Open();

}

private void guna2GradientButton2\_Click(object sender, EventArgs e)

{

mainForm.StudentRegister\_Open();

}

}

}

**Result**

namespace Test\_tuzish

{

public partial class Result : UserControl

{

private MainForm mainForm;

public Result(MainForm mainForm, string name, string time1, int ball1, int questions\_count)

{

this.mainForm = mainForm;

InitializeComponent();

fish.Text += name;

time.Text += time1;

ball.Text += ball1+" / "+ questions\_count;

double persent = ((double)ball1 / questions\_count) \* 100;

guna2CircleProgressBar1.Value = (int)persent;

if(persent > 60)

{

guna2CircleProgressBar1.ProgressColor = Color.Green;

guna2CircleProgressBar1.ProgressColor2 = Color.GreenYellow;

}

else

{

guna2CircleProgressBar1.ProgressColor = Color.Maroon;

guna2CircleProgressBar1.ProgressColor2 = Color.Red;

}

}

private void endButton\_Click(object sender, EventArgs e)

{

mainForm.Menu\_Open();

}

}

}

**Student**

namespace Test\_tuzish

{

public partial class Student : UserControl

{

Query q = new Query();

private MainForm mainForm;

private string student\_name;

private string student\_group;

private string test\_name;

private int questionsCount;

private int timeLeft;

private Timer timer;

public Student(MainForm mainForm, string student\_name, string student\_group, string test\_name, int questionsCount, int timeLeft)

{

this.mainForm = mainForm;

this.student\_name = student\_name.Replace("'", "`");

this.student\_group = student\_group;

this.test\_name = test\_name;

this.questionsCount = questionsCount;

this.timeLeft = timeLeft;

InitializeComponent();

LoadQuestions();

}

private Dictionary<string, (string TrueAnswer, List<string> answers)> questionsDict;

private List<string> selectedQuestions;

private Dictionary<string, int> keyValuePairs = new Dictionary<string, int>();

private int studentMark = 0;

private string selectedAnswer = "";

private int currentQuestionIndex;

private void LoadQuestions()

{

string filePath = TxtReader.GetPathOfTest(test\_name);

questionsDict = TxtReader.GetQuestionsDic(filePath);

var random = new Random();

selectedQuestions = questionsDict.Keys.ToList();

selectedQuestions = selectedQuestions.OrderBy(x => random.Next()).Take(questionsCount).ToList();

currentQuestionIndex = 0;

ShowQuestion(selectedQuestions[currentQuestionIndex]);

timer = new Timer();

timer.Interval = 1000; // Интервал таймера в миллисекундах (1000 мс = 1 сек)

timer.Tick += Timer\_Tick;

timer.Start();

}

private void ShowQuestion(string question)

{

selectedAnswer = "";

questionRichtextbox.Text = question;

var answers = questionsDict[question].answers;

answerA\_text.Text = answers[0];

answerB\_text.Text = answers[1];

answerC\_text.Text = answers[2];

answerD\_text.Text = answers[3];

A\_answer.Checked = B\_answer.Checked = C\_answer.Checked = D\_answer.Checked = false;

questionNumber.Text = (currentQuestionIndex + 1).ToString();

questionNumber.Location = new Point((panel1.Width - questionNumber.Width) / 2, (panel1.Height - questionNumber.Height) / 2);

}

private void Check(bool forward)

{

if (keyValuePairs.Keys.Count != questionsCount)

{

if (keyValuePairs.Values.Contains(currentQuestionIndex + 1))

{

if (forward)

{

currentQuestionIndex++;

if (currentQuestionIndex == questionsCount)

{

currentQuestionIndex = 0;

}

Check(true);

}

else

{

currentQuestionIndex--;

if (currentQuestionIndex == -1)

{

currentQuestionIndex = questionsCount-1;

}

Check(false);

}

}

else

{

ShowQuestion(selectedQuestions[currentQuestionIndex]);

}

}

else

{

The\_End();

}

}

private void Comfirm(object sender, EventArgs e)

{

if (selectedAnswer != "")

{

if (keyValuePairs.Keys.Count != questionsCount)

{

if (!keyValuePairs.Keys.Contains(selectedQuestions[currentQuestionIndex]))

{

keyValuePairs.Add(selectedQuestions[currentQuestionIndex], currentQuestionIndex + 1);

}

var correctAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].TrueAnswer;

if (correctAnswer == selectedAnswer)

{

currentQuestionIndex++;

if (currentQuestionIndex == questionsCount)

{

currentQuestionIndex = 0;

}

studentMark++;

Check(true);

}

else

{

currentQuestionIndex++;

if (currentQuestionIndex == questionsCount)

{

currentQuestionIndex = 0;

}

Check(true);

}

}

else

{

endButton\_Click(sender, e);

}

}

else

{

Next(sender, e);

}

}

private void Next(object sender, EventArgs e)

{

currentQuestionIndex++;

if (currentQuestionIndex == questionsCount)

{

currentQuestionIndex = 0;

}

Check(true);

}

private void Preview(object sender, EventArgs e)

{

currentQuestionIndex--;

if (currentQuestionIndex == -1)

{

currentQuestionIndex = questionsCount-1;

}

Check(false);

}

private void answerA\_text\_Click(object sender, EventArgs e)

{

A\_answer.Checked = true;

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[0];

}

private void answerB\_text\_Click(object sender, EventArgs e)

{

B\_answer.Checked = true;

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[1];

}

private void answerC\_text\_Click(object sender, EventArgs e)

{

C\_answer.Checked = true;

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[2];

}

private void answerD\_text\_Click(object sender, EventArgs e)

{

D\_answer.Checked = true;

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[3];

}

private int seconds = 0;

private void Timer\_Tick(object sender, EventArgs e)

{

seconds--;

if (seconds <= 0)

{

seconds = 59;

timeLeft--;

}

if (timeLeft <= 0)

{

mainForm.Result\_Open(student\_name, "00:00", studentMark, questionsCount);

timer.Stop();

}

else

{

timeLeftLABEL.Text = $"{timeLeft}:{seconds}";

}

}

private void endButton\_Click(object sender, EventArgs e)

{

The\_End();

}

public void The\_End()

{

if (keyValuePairs.Keys.Count == questionsCount)

{

timer.Stop();

q.Insert($"INSERT INTO RESULTS VALUES('{student\_name}','{student\_group}','{test\_name}','{studentMark}/{questionsCount}','{DateTime.Now.ToString("dd.MM.yyyy HH:mm")}')");

mainForm.Result\_Open(student\_name, $"{timeLeft}:{seconds}", studentMark, questionsCount);

}

else

{

if (MessageBox.Show("Yakunlamoqchimisiz?", "Yakunlash", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

timer.Stop();

q.Insert($"INSERT INTO RESULTS VALUES('{student\_name}','{student\_group}','{test\_name}','{studentMark}/{questionsCount}','{DateTime.Now.ToString("dd.MM.yyyy HH:mm")}')");

mainForm.Result\_Open(student\_name, $"{timeLeft}:{seconds}", studentMark, questionsCount);

}

}

}

private void A\_answer\_CheckedChanged(object sender, EventArgs e)

{

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[0];

}

private void B\_answer\_CheckedChanged(object sender, EventArgs e)

{

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[1];

}

private void C\_answer\_CheckedChanged(object sender, EventArgs e)

{

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[2];

}

private void D\_answer\_CheckedChanged(object sender, EventArgs e)

{

selectedAnswer = questionsDict[selectedQuestions[currentQuestionIndex]].answers[3];

}

}

}

**StudentRegistration**

namespace Test\_tuzish

{

public partial class StudentRegistration : UserControl

{

Query q = new Query();

private MainForm mainForm;

int timeLeft = 0;

int questionsCount = 0;

public StudentRegistration(MainForm mainForm)

{

this.mainForm = mainForm;

InitializeComponent();

q.FillComboBox("SUBJECT", "TESTS", subjectCombobox);

}

private void subjectCombobox\_SelectedIndexChanged(object sender, EventArgs e)

{

var table = q.Select($"SELECT COUNT,TIME FROM TESTS WHERE SUBJECT = '{subjectCombobox.SelectedItem}'");

questionsCount = int.Parse(table.Rows[0][0].ToString());

timeLeft = int.Parse(table.Rows[0][1].ToString());

questionsCountLabel.Text = questionsCount + "ta savol";

timeLeftLabel.Text = timeLeft + " daqiqa";

}

private void guna2GradientButton2\_Click(object sender, EventArgs e)

{

if (subjectCombobox.SelectedIndex == -1)

{

subjectCombobox.Focus();

errorLabel.Visible = true;

}

else if (string.IsNullOrWhiteSpace(studentNameTxtbox.Text))

{

studentNameTxtbox.Focus();

errorLabel.Visible = true;

}

else if (string.IsNullOrWhiteSpace(studentGroupTxtBox.Text))

{

studentGroupTxtBox.Focus();

errorLabel.Visible = true;

}

else

{

mainForm.Student\_Open(studentNameTxtbox.Text,studentGroupTxtBox.Text,subjectCombobox.SelectedItem.ToString(), questionsCount, timeLeft);

}

}

private void exitBtn2\_Click(object sender, EventArgs e)

{

mainForm.Menu\_Open();

}

}

}