**Millionaire Quiz**

Windows Forms Project:

* Мартин Розин Сопков 223222
* Никола Серафимов 223125
* Христијан Стоимилов 223163

**1. Опис на апликацијата**

**„Millionaire Quiz“** е едукативно-забавна апликација развиена како **Windows Forms** игра со цел да се симулира квизот од популарната ТВ-емисија “*Who Wants to Be a Millionaire?”*. Апликацијата служи како интерактивен начин за тестирање и унапредување на знаењето на корисникот преку забавен формат со прашања и награди.

Главната функционалност е едноставна: играчот добива 15 прашања со растечка тежина, поделени во три нивоа: **Easy, Medium и Hard**. Со секој точен одговор, се зголемува виртуелната добивка, а крајната цел е да се достигне износ од **1.000.000 евра**.

Прашањата се структурирани, така што секое следно е покомплексно од претходното, што создава предизвик за играчот. Овој постепен напредок го одржува интересот до крај и го стимулира логичкото размислување и концентрацијата.

Во текот на играта, корисникот има на располагање **три типови на помош (lifelines)**, кои можат да му помогнат при одговор на потешките прашања:

* **50/50** – Елеминира два неточни одговори.
* **Call a Friend** – Симулира телефонски повик со пријател кој дава можен одговор.
* **Ask the Audience** – Симулира повик со публиката кој дава можен одговор

Секој од овие помошни алати може да се искористи само еднаш, што му дава тактичка компонента на играта – играчот треба мудро да одлучи кога и како ќе ги употреби.

Апликацијата е наменета за сите возрасни групи и може да се користи како алатка за учење, забава или пријателско натпреварување.

**2. Решение на проблемот**

Апликацијата е развиена во **C# со .NET Framework**, користејќи **Windows Forms** за графичкиот кориснички интерфејс. Структурата на проектот е поделена во неколку класи и форми, кои го организираат кодот на логичен и модуларен начин.

**Чување и организација на податоци:**

* **Прашањата** се зачувани како објекти од класа Question, која ги содржи следните атрибути:
  + **Question** – текстот на прашањето
  + **Answers []** – низа од 4 одговори
  + **Correct** – точен одговор
  + **Difficulty** – ниво на тежина (Easy, Medium, Hard)

**3. Опис на функција/класа**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Што прави оваа функција?**

* **button5\_Click е** метод поврзан со клик на копчиња за помош:
  + **callFriendButton** (Телефонирај пријател)
  + **askAudienceButton** (Прашaј ја публиката)
* При повик, се симулира одговор од пријателот или публиката, кој има одредена шанса да го каже точниот одговор.

**Детали:**

* **chance** е веројатноста пријателот/публиката да даде точен одговор:
  + За лесни прашања: 90% точност
  + За средни: 70%
  + За тешки: 50%
* Се прави случаен број **(rand.NextDouble())** и ако е помал или еднаков на **chance**, се одбира точниот одговор.
* Ако не, се избира случаен погрешен одговор од останатите три опции.
* Потоа, според тоа кое копче е кликнато (**callFriendButton** или **askAudienceButton**), се прикажува порака со предлогот.
* Копчето се оневозможува по една употреба (**clickedButton.Enabled = false;**).

**4. Скриншоти и кратко упатство за користење**

**Упатство:**

1. При стартување на апликацијата, веднаш се започнува со квизот, корисникот добива прво прашање и 4 понудени одговори.
2. За секое прашање:

* Корисникот избира одговор.
* Ако е точен, се продолжува понатаму, ако одговорот е погрешен играта завршува.
* Достапни се **3 помошни опции** кои може да се користат по желба, само еднаш во текот на целата игра.

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**5. Опис на користење на вештачка интелигенција Google Gemini**

Класата GenerateQuestion е задолжена за автоматско испраќање на барање до **вештачка интелигенција (AI)**, со цел генерирање на ново прашање. При тоа, се задава ниво на тежина (Easy, Medium или Hard) и се избира случајна категорија, како што се историја, наука, географија и слично.

**Како функционира:**

**1. Подесување на API повикот**

* Користи **Google Gemini API**.
* ApiKey: Овозможува пристап до Gemini моделот (твоја API лозинка).
* Endpoint: URL адреса преку која праќаме барање до Gemini моделот.

**2. Генерирање на барањето**

* Избира случајна категорија (на пр. Geography).
* Се креира **prompt**: текст со упатства до вештачката интелигенција.  
   Моделот треба да врати само **1 прашање**, со 4 одговори, еден точен.  
   Строго структуриран формат.

**3. Креирање JSON барање**

* Податоците се спакувани во JSON формат.
* Параметрите temperature, topK, topP ја контролираат **креативноста** на моделот.

**4. Повик до Gemini API**

* Го испраќа барањето.
* Чека одговор од моделот.

**5. Парсирање на одговорот**

* Прочитува одговорот од API.
* Извлекува само генерираниот текст од вештачката интелигенција (прашањето).

**6. Грешки и пораки**

* Ако API не врати валиден одговор - се прикажува **MessageBox** со грешка.
* Ако текстот е празен - се прикажува **предупредување**.