

Rapport de projet d'implémentation du modèle d'architecture MVC

RAZAFINDRAIBE Hery Jhonny

Avril 2024

"QUIZ WEBSITE"



Prof: Jabrane Jihane

Ecole National de Sciences Appliquées de Tanger

Sommaire

01 | Introduction

02 | Contexte du projet

03 | Vue d'ensemble du modèle MVC

04 | Communication entre les couches

05 | Tests et validation

06 | Conclusion

Introduction

1

A-Qu'est ce que le modèle MVC ?

Le Modèle-Vue-Contrôleur (MVC) est un motif de conception largement utilisé dans le développement logiciel pour organiser et structurer les applications.

Il divise une application en trois composants principaux :

- le modèle
- la vue
- le contrôleur

Le modèle : il représente les données de l'application. Il définit aussi l'interaction avec la base de données et le traitement de ces données

La vue: elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente d'afficher les données que lui fournit le modèle

Le Contrôleur : il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

Introduction

2

Les environnements de développement que j'ai utilisé pour implémenter ce modèle

1

Visual Studio

Il permet de gérer le développement d'applications .NET telles que celles basées sur ASP.NET Core.

2

Visual Studio Code

VS Code est un éditeur de code léger, rapide et que j'ai déjà utilisé depuis longtemps

3

ASP.NET Core

Il offre des fonctionnalités telles que la gestion des requêtes HTTP, la configuration, la sécurité, et la prise en charge de l'architecture MVC, ce qui en fait un choix idéal pour développer des applications web robustes comme un système de quiz.

Introduction

3

Les environnements de développement que j'ai utilisé pour implémenter ce modèle

4

React Js

React est une bibliothèque JavaScript populaire pour la création d'interfaces utilisateur interactives. Je l'utilise pour gérer la partie Vue de mon QUIZ

5

SQL

Il est utilisé dans mon cas pour stocker et gérer les questions du quiz, les réponses des utilisateurs, les scores, etc.,

6

Langage C sharp

Il est bien adapté au développement d'applications web grâce à ASP.NET Core, offrant une performance élevée, une sécurité renforcée et une large gamme de bibliothèques et de frameworks.

Contexte du projet

4

Description du Quiz

Le projet que j'ai choisi pour implémenter le modèle MVC c'est un **QUIZ**.
Vous vous demandez peut-être pourquoi ? Eh bien, la réponse c'est que :

- Un Quiz implique la gestion de différentes parties : **les questions, les réponses, les scores, l'interface utilisateur pour afficher les questions et recueillir les réponses**, etc...
- En utilisant le modèle MVC, je pouvais séparer clairement ces différentes préoccupations :
 1. Le modèle stocke les questions ainsi que les réponses du QUIZ et les informations sur le participant dans une base de donnée SQL.

Table stocker les questions et les réponses ainsi que les options:

Résultats		Messages							
QnId	QnWords	ImageName	Option1	Option2	Option3	Option4	Answer		
1	1	What does HTML stand for?	NULL	Hyper Trainer Marking Language	Hyper Text Marketing Language	Hyper Text Markup Language	Hyper Text Markup Leveler	2	
2	2	The brain of any computer system is	NULL	ALU	Memory	CPU	Control unit	2	
3	3	Which of the following computer language is used...	NULL	FORTTRAN	PROLOG	C	COBOL	1	
4	4	What is the primary requisite of a good computer ...	NULL	Mathematical Mind	Artistic mind	Logical Mind	Scientific Knowledge	2	
5	5	Name the device.	mouse.png	Keyboard	Monitor	Mouse	Graphics Card	2	
6	6	The first mechanical computer designed by Charl...	NULL	Analytical Engine	Calculator	Processor	Abacus	0	
7	7	One byte is equivalent to ?	NULL	4 bits	8 bits	16 bits	32 bits	1	
8	8	Web pages are written using ?	NULL	FTP	UML	HTML	URL	2	
9	9	Which of the following is NOT operating system ?	NULL	Dos	Unix	Window NT	CSS	3	
10	10	What is the full form of IP ?	NULL	Interface Program	Interface Protocol	Internet program	Internet Protocol	3	

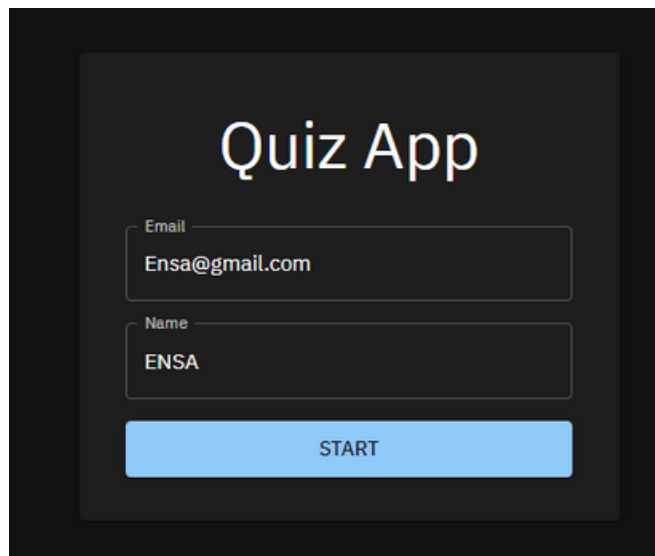
Table pour stocker les informations sur le participant:

MALASO\SQLEXPRESS...dbo.Participants		SQLQuery1.sql - (I...(MALASO\User (65))	
Nom de la colonne		Type de données	Autoriser les vale...
ParticipantId		int	<input type="checkbox"/>
Email		nvarchar(50)	<input type="checkbox"/>
Name		nvarchar(50)	<input type="checkbox"/>
Score		int	<input type="checkbox"/>
TimeTaken		int	<input type="checkbox"/>

Contexte du projet

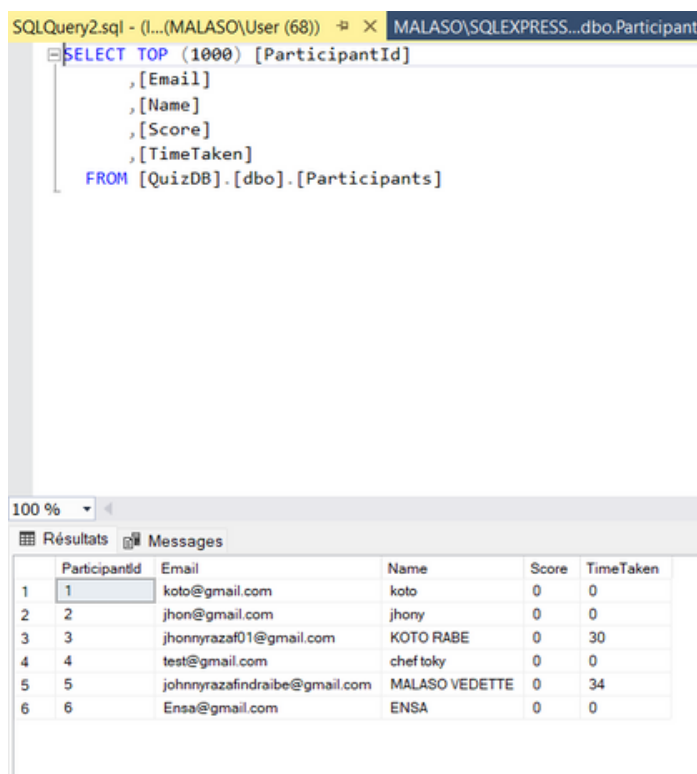
5

On va saisir un participant



A dark-themed login form titled "Quiz App". It features two input fields: "Email" with the text "Ensa@gmail.com" and "Name" with the text "ENSA". Below the fields is a blue "START" button.

En cliquant sur start, ces informations seront directement stocker dans son table



SQLQuery2.sql - (L... (MALASO\User (68))) MALASO\SQLEXPRESS...dbo.Participant

```
SELECT TOP (1000) [ParticipantId]  
      ,[Email]  
      ,[Name]  
      ,[Score]  
      ,[TimeTaken]  
FROM [QuizDB].[dbo].[Participants]
```

100 %

Résultats Messages

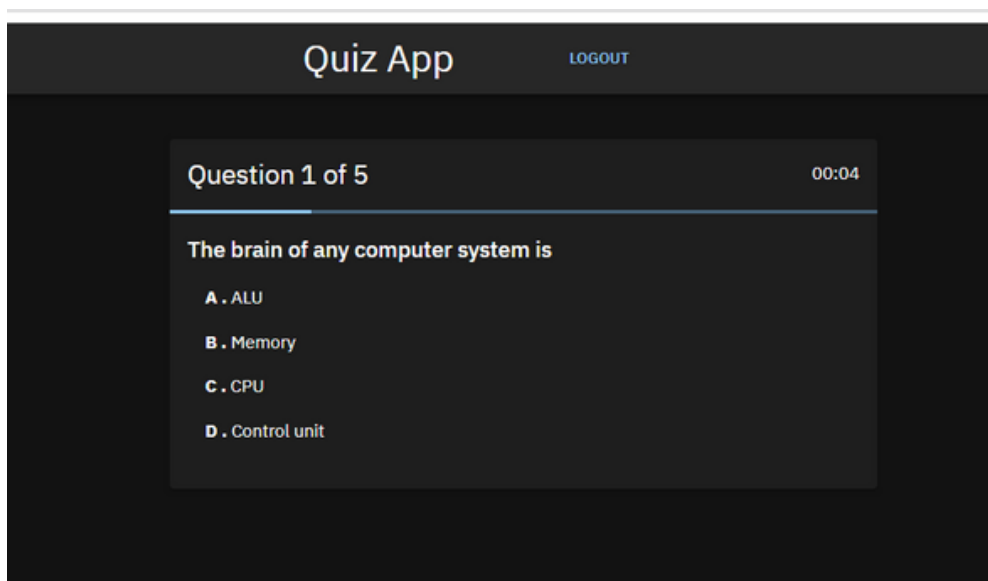
	ParticipantId	Email	Name	Score	TimeTaken
1	1	koto@gmail.com	koto	0	0
2	2	jhon@gmail.com	jhony	0	0
3	3	jhonnryazaf01@gmail.com	KOTO RABE	0	30
4	4	test@gmail.com	chef toky	0	0
5	5	johnnyrazafindraibe@gmail.com	MALASO VEDETTE	0	34
6	6	Ensa@gmail.com	ENSA	0	0

Eh voilà, le tour est joué, notre utilisateur est insérer à la dernière ligne

Contexte du projet

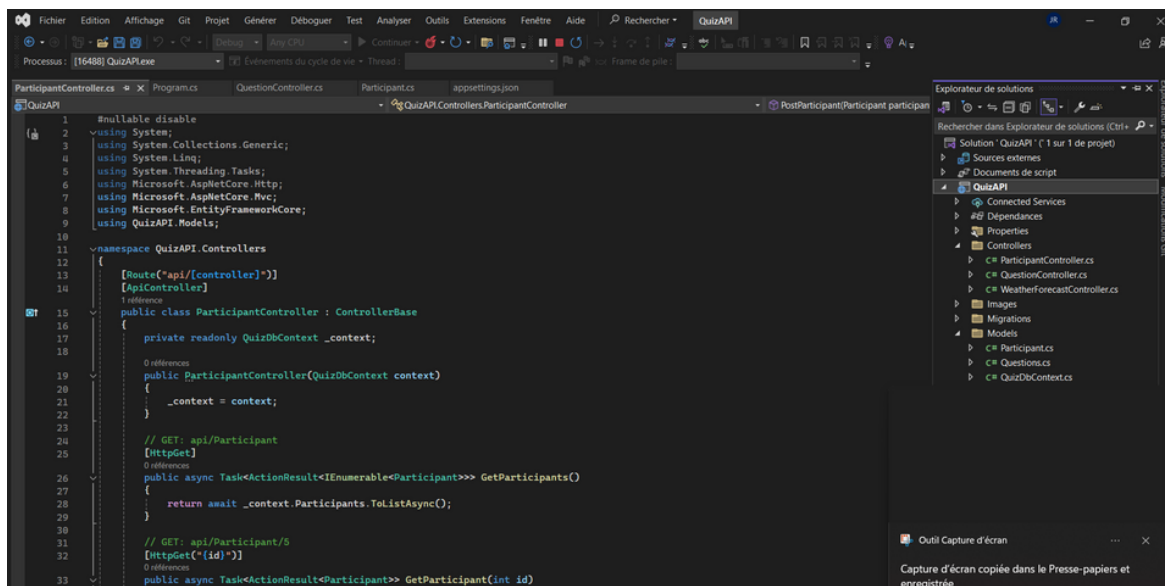
6

2-La vue affiche les questions et permet à l'utilisateur de choisir une réponse



3-Le contrôleur gère les interactions entre le modèle et la vue, sachant que j'ai utiliser 2 Contrôleurs :

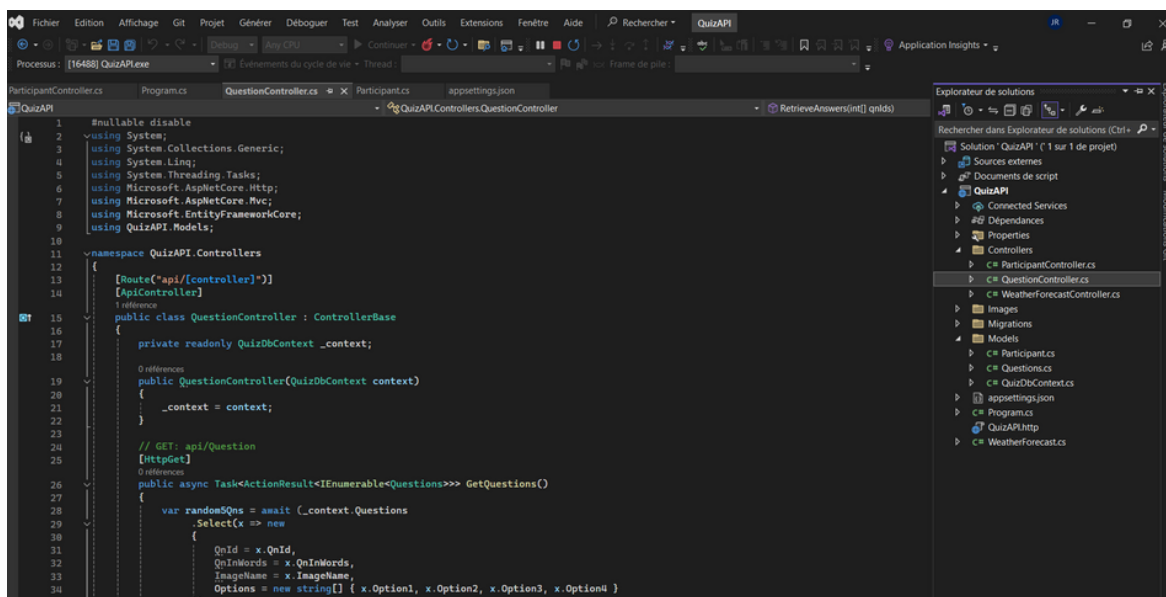
- Un contrôleur qui écoute les réponses de l'Utilisateur à travers un port (localhost:3000) que j'ai nommé "ParticipantController" et les comparent avec la réponse dans la base de donnée



Contexte du projet

7

- Un contrôleur qui récupère aléatoirement 5 questions parmi 10 dans la base des données et l'affiche dans la partie Vue



```
1 #nullable disable
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Threading.Tasks;
6 using Microsoft.AspNetCore.Http;
7 using Microsoft.AspNetCore.Mvc;
8 using Microsoft.EntityFrameworkCore;
9 using QuizAPI.Models;
10
11 namespace QuizAPI.Controllers
12 {
13     [Route("api/[controller]")]
14     [ApiController]
15     public class QuestionController : ControllerBase
16     {
17         private readonly QuizDbContext _context;
18
19         public QuestionController(QuizDbContext context)
20         {
21             _context = context;
22         }
23
24         // GET: api/question
25         [HttpGet]
26         public async Task<ActionResult<IEnumerable<Questions>>> GetQuestions()
27         {
28             var randomQns = await _context.Questions
29                 .Select(x => new
30                 {
31                     QnId = x.QnId,
32                     QnInWords = x.QnInWords,
33                     ImageName = x.ImageName,
34                     Options = new string[] { x.Option1, x.Option2, x.Option3, x.Option4 }
35                 })
36                 .Take(5)
37                 .ToListAsync();
38             return Ok(randomQns);
39         }
40     }
41 }
```

Je mets à votre disposition mon code source afin que vous puissiez vérifier tout les codes du contrôleur (en C#) , sachant que ce captures sont juste une partie du code

>>>> Afin de bien compiler le code, il est primordial que vous installer certaines framework
Material UI pour React: **npm install @mui/material @emotion/react @emotion/styled**

Communication entre les Couches

8

1-Interaction avec le Modèle (QuestionsContrôleur) :

- Lorsqu'un utilisateur interagit avec l'interface utilisateur (Vue) pour répondre à une question, l'action est capturée par le QuestionsContrôleur.
- Le QuestionsContrôleur utilise le modèle (Modèle) pour récupérer la question actuelle à partir de la base de données. Cette étape implique généralement un appel à une méthode du modèle qui récupère la question en fonction de son ID ou de sa position dans le Quiz.
- Une fois que la question est récupérée du modèle, le QuestionsContrôleur la transmet à la Vue pour affichage à l'utilisateur.

2.Traitement des Réponses (ParticipantContrôleur) :

- Lorsque l'utilisateur soumet sa réponse à une question, l'action est gérée par le ParticipantContrôleur.
- Le ParticipantContrôleur récupère la réponse fournie par l'utilisateur depuis l'interface utilisateur (Vue).
- Ensuite, le ParticipantContrôleur utilise le modèle (Modèle) pour comparer la réponse donnée par l'utilisateur avec la réponse correcte enregistrée dans la base de données. Cela implique typiquement un appel à une méthode du modèle qui effectue cette comparaison et retourne le résultat (correct ou incorrect).

3-Transmission des Résultats à la Vue :

- Une fois que le modèle (Modèle) a évalué la réponse de l'utilisateur, le ParticipantContrôleur reçoit le résultat de cette évaluation (par exemple, "correct" ou "incorrect").
- Le ParticipantContrôleur transmet ensuite ce résultat à l'interface utilisateur (Vue) pour affichage à l'utilisateur. Cela peut impliquer la modification de l'état de la Vue pour indiquer à l'utilisateur si sa réponse était correcte ou incorrecte.

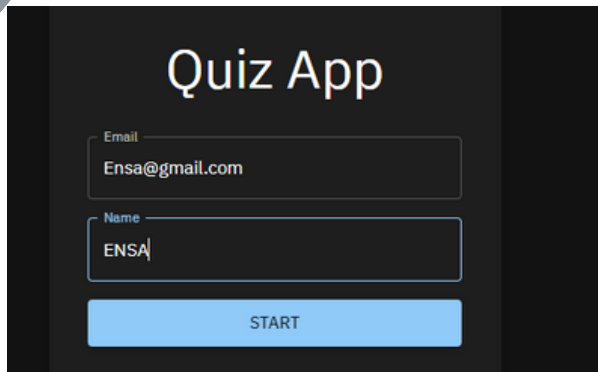
Résumé de Flux d'Interaction :

- 1.L'utilisateur visualise la question actuelle sur l'interface utilisateur (Vue), qui est chargée par le QuestionsContrôleur.
- 2.L'utilisateur sélectionne une réponse à la question et la soumet.
- 3.L'action de soumission est capturée par le ParticipantContrôleur.
- 4.Le ParticipantContrôleur récupère la réponse de l'utilisateur et utilise le modèle (Modèle) pour comparer cette réponse avec la réponse correcte enregistrée dans la base de données.
- 5.Le résultat de cette comparaison est transmis par le ParticipantContrôleur à l'interface utilisateur (Vue) pour afficher le feedback à l'utilisateur (par exemple, "Bonne réponse !" ou "Désolé, ce n'est pas correct.").

Test et Validation

9

On va prendre l'utilisateur qu'on a saisi ci-dessus : ENSA@gmail.com

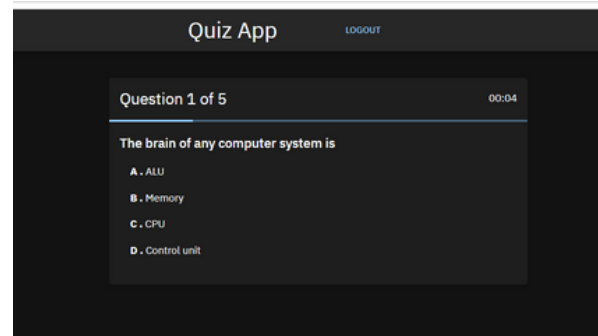


Quiz App

Email
Ensa@gmail.com

Name
ENSA

START

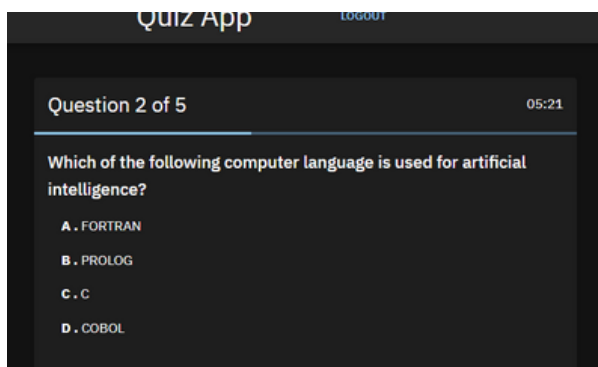


Quiz App

Question 1 of 5 00:04

The brain of any computer system is

- A . ALU
- B . Memory
- C . CPU
- D . Control unit

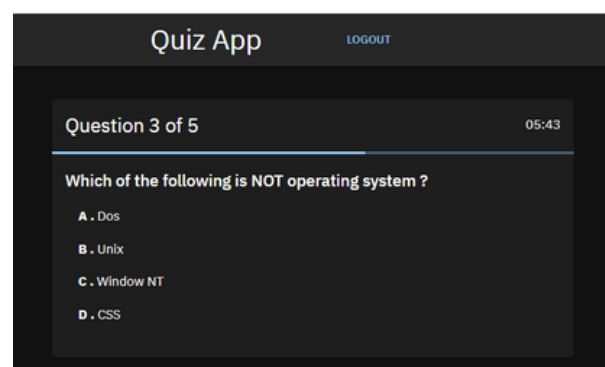


Quiz App

Question 2 of 5 05:21

Which of the following computer language is used for artificial intelligence?

- A . FORTRAN
- B . PROLOG
- C . C
- D . COBOL

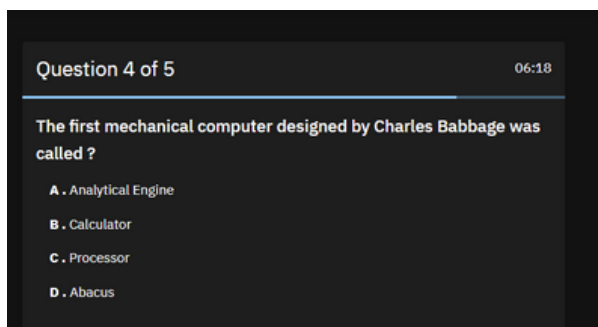


Quiz App

Question 3 of 5 05:43

Which of the following is NOT operating system ?

- A . Dos
- B . Unix
- C . Window NT
- D . CSS

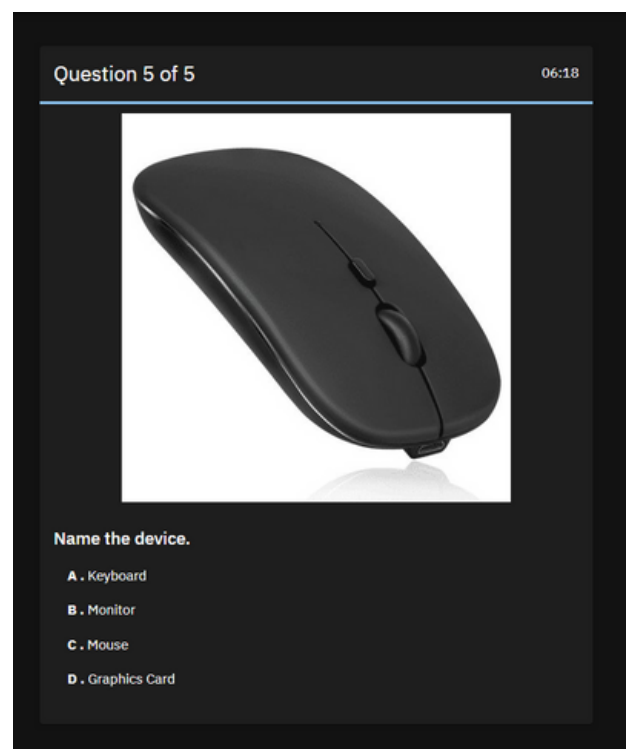


Quiz App

Question 4 of 5 06:18


The first mechanical computer designed by Charles Babbage was called ?

- A . Analytical Engine
- B . Calculator
- C . Processor
- D . Abacus



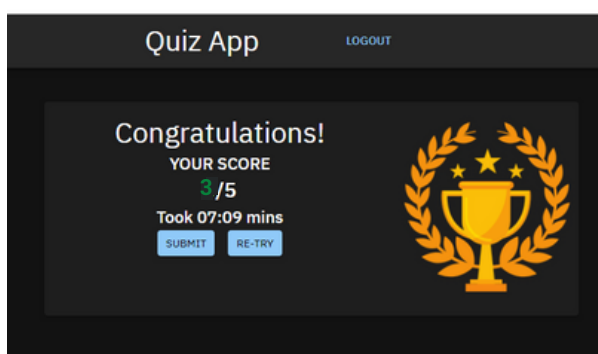
Quiz App

Question 5 of 5 06:18



Name the device.

- A . Keyboard
- B . Monitor
- C . Mouse
- D . Graphics Card




Quiz App

Congratulations!

YOUR SCORE
3/5

Took 07:09 mins

SUBMIT RE-TRY



Conclusion

En travaillant sur cette implémentation du modèle MVC pour mon projet de Quiz, j'ai pu apprécier pleinement les avantages de cette architecture logicielle. La séparation claire entre le Modèle, la Vue et le Contrôleur m'a permis de développer une application bien organisée et maintenable.

L'un des principaux avantages que j'ai constatés est la modularité du code. En divisant mon application en différents composants selon le modèle MVC, j'ai pu réutiliser efficacement le code pour gérer les questions, les réponses et les interactions avec l'utilisateur. Cela m'a permis de gagner du temps et d'assurer la cohérence dans le développement de mon Quiz.

Par ailleurs, la structure MVC a facilité les tests unitaires et le débogage. En isolant chaque composant, j'ai pu identifier et corriger les problèmes plus rapidement. Cela a renforcé la fiabilité de mon application et m'a donné plus de confiance dans sa qualité.

Travailler sur ce projet m'a également permis de prendre des décisions de conception de manière autonome, ce qui a été à la fois gratifiant et éducatif. J'ai pu voir directement les résultats de mes choix architecturaux et en tirer des leçons précieuses pour mes futurs projets.

En résumé, l'implémentation du modèle MVC a été une expérience enrichissante. Cela m'a non seulement permis de créer une application fonctionnelle et convaincante, mais aussi d'approfondir ma compréhension des bonnes pratiques de développement logiciel. Je recommande vivement l'utilisation du modèle MVC, pour sa capacité à rationaliser le processus de développement et à produire un code robuste et maintenable



Remerciements

Je tiens à exprimer mes sincères remerciements à **Jabrane Jihane** pour votre soutien et vos éducations tout au long du demi semestre S3 .

En particulier, je suis reconnaissant pour :

- Votre cour bien détaillé qui m'a aidé à comprendre les concepts clés du modèle MVC et leur application pratique.
- Votre disponibilité et votre rigueur pour répondre à nos questions et résoudre nos problèmes tout au long du TP Swift à l'école.

Ce projet a été une occasion précieuse d'apprendre et de grandir en tant que développeur, et cela n'aurait pas été possible sans votre initiative d'inclure de TP dans la notation de votre matière.

Je tiens également à remercier l'ensemble de l'équipe pédagogique pour leur engagement envers notre réussite académique et professionnelle.