

RÉSUMÉ DU PROJET - SITE DE QUIZ

1. *index.php* (Page d'accueil)

- **Rôle** : Point d'entrée du site
- **Contenu** : Formulaire de connexion + lien vers inscription
- **Fonctionnalités** : Redirection selon le rôle (prof/élève)

2. *register.php*

- **Rôle** : Inscription des nouveaux utilisateurs
- **Contenu** : Formulaire (nom, email, mot de passe, rôle)
- **Sécurité** : Validation des données, hachage mot de passe

3. *login.php*

- **Rôle** : Connexion des utilisateurs
- **Contenu** : Email + mot de passe
- **Redirection** : Vers tableau de bord selon le rôle

INTERFACE PROFESSEUR

4. *prof-dashboard.php*

- **Rôle** : Tableau de bord principal du professeur
- **Contenu** :
 - Liste de tous ses quiz créés
 - Statistiques (nombre de quiz, tentatives, scores moyens)
 - Boutons d'actions (créer, modifier, voir résultats)
 - Dernières tentatives des étudiants

5. *create-quiz.php*

- **Rôle** : Création de nouveaux quiz
- **Contenu** :
 - Informations générales (titre, description, matière)
 - Timer personnalisable (1-60 minutes)
 - Ajout dynamique de questions QCM (max 25)
 - 4 options par question + bonne réponse
- **JavaScript** : Ajout/suppression questions en temps réel

6. *edit-quiz.php*

- **Rôle** : Modification des quiz existants
- **Contenu** :
 - Formulaire pré-rempli avec données actuelles
 - Modification des questions existantes
 - Zone de suppression définitive du quiz
- **Sécurité** : Vérification que le quiz appartient au professeur

INTERFACE ÉLÈVE

7. *student-dashboard.php*

- **Rôle** : Tableau de bord de l'élève
- **Contenu** :
 - Liste des quiz disponibles par matière
 - Filtrage par matière (Maths, Histoire, Anglais)
 - Informations quiz (durée, nb questions, professeur)
 - Historique des derniers résultats personnels

8. *take-quiz.php*

- **Rôle** : Interface de passage du quiz
- **Contenu** :
 - **Timer en temps réel** avec alertes visuelles
 - Navigation question par question
 - Barre de progression

- Options A, B, C, D pour chaque question
- **Anti-triche :**
 - Soumission automatique à la fin du timer
 - Protection contre fermeture accidentelle
 - Une seule tentative par élève

PAGES DE RÉSULTATS

9. quiz-result.php

- **Rôle :** Affichage des résultats après quiz
- **Contenu :**
 - Score final en pourcentage
 - Détail question par question
 - Réponses correctes vs données
 - Temps total utilisé
 - Bouton retour au dashboard

10. results.php

- **Rôle :** Historique complet des résultats
- **Pour élève :** Ses propres tentatives
- **Pour professeur :** Résultats de tous les étudiants

FICHIERS TECHNIQUES

config/database.php

- Connexion PDO à MySQL
- Configuration XAMPP (localhost, root, pas de mot de passe)

includes/functions.php

- Fonctions utilitaires (authentification, sécurité, validation)
- Gestion des sessions et redirections

BASE DE DONNÉES (6 TABLES)

1. **users** : Utilisateurs (prof/élève)
2. **subjects** : Matières (Maths, Histoire, Anglais)
3. **quizzes** : Informations des quiz
4. **questions** : Questions avec 4 options
5. **quiz_attempts** : Tentatives des étudiants
6. **student_answers** : Réponses détaillées

DESIGN & EXPÉRIENCE UTILISATEUR

- **Style moderne** avec dégradés et animations CSS
- **Interface responsive** pour mobile/desktop
- **Navigation intuitive** avec breadcrumbs
- **Feedback visuel** (alertes, confirmations)
- **Timer visuel** avec changements de couleur

MESURES ANTI-TRICHE

- ✓ **Timer personnalisé** (1-60 minutes)
- ✓ **Soumission automatique** à la fin
- ✓ **Une tentative unique** par élève
- ✓ **Questions mélangées** (ORDER BY RAND())
- ✓ **Protection session** et validation serveur

TECHNOLOGIES UTILISÉES

- **Backend** : PHP pur avec PDO
- **Frontend** : HTML5, CSS3, JavaScript
- **Base de données** : MySQL
- **Serveur local** : XAMPP
- **Design** : CSS Grid/Flexbox + animations

EXPLICATION TECHNIQUE - CSS & ANIMATIONS

ANIMATIONS CSS UTILISÉES

1. Animation d'entrée (fadeInUp)

CSS

```
@keyframes fadeInUp {  
  from {  
    opacity: 0;  
    transform: translateY(30px);  
  }  
  to {  
    opacity: 1;  
    transform: translateY(0);  
  }  
}
```

Objectif : Les éléments apparaissent en "glissant" du bas vers le haut **Application** : Sur les cartes de quiz, sections principales **Durée** : 1s à 1.8s selon l'élément

2. Animation de titre (bounce)

CSS

```
@keyframes bounce {  
  0%, 20%, 50%, 80%, 100% { transform: translateY(0); }  
  40% { transform: translateY(-10px); }  
  60% { transform: translateY(-5px); }  
}
```

Objectif : Effet de rebond sur les titres principaux **Application** : Titres h1 des pages

Répétition : Infinite pour attirer l'attention

3. Animation pulsante (pulse)

CSS

```
@keyframes pulse {  
  0%, 100% { transform: scale(1); }  
  50% { transform: scale(1.05); }  
}
```

Objectif : Effet de respiration sur les éléments importants **Application** : Logo, timer en danger, boutons d'action

EFFETS DE SURVOL (HOVER)

1. Effet de profondeur sur les cartes

```
CSS  
.quiz-card:hover {  
  transform: translateY(-10px);  
  box-shadow: 0 20px 40px rgba(0,0,0,0.2);  
}
```

Résultat : Les cartes "s'élèvent" au survol

2. Effet d'ondulation sur les boutons

```
CSS  
.btn:before {  
  content: '';  
  position: absolute;  
  background: rgba(255, 255, 255, 0.3);  
  border-radius: 50%;  
  transition: width 0.6s, height 0.6s;  
}  
.btn:hover:before {  
  width: 300px;  
  height: 300px;  
}
```

Résultat : Ondulation blanche qui s'étend au clic

DESIGN VISUEL

1. Dégradés de fond

CSS

```
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
```

Choix : Bleu vers violet pour un aspect moderne et professionnel

2. Effet de flou (backdrop-filter)

CSS

```
.header {  
    background: rgba(255, 255, 255, 0.1);  
    backdrop-filter: blur(10px);  
}
```

Résultat : Header semi-transparent avec flou d'arrière-plan

3. Glassmorphism

CSS

```
background: rgba(255, 255, 255, 0.95);  
box-shadow: 0 20px 40px rgba(0,0,0,0.1);
```

Style : Cartes semi-transparentes avec ombres douces

JAVASCRIPT FONCTIONNEL

1. Timer en temps réel (take-quiz.php)

javascript

```
function startTimer() {  
    timerInterval = setInterval(function() {  
        timeLeft--;
```

```

        updateTimerDisplay();

        if (timeLeft <= 0) {
            clearInterval(timerInterval);
            document.getElementById('quizForm').submit();
        }
    }, 1000);
}

```

Fonctionnalités :

- Décompte en temps réel
- Changement de couleur selon le temps restant
- Soumission automatique à zéro

2. Navigation dynamique des questions

javascript

```

function changeQuestion(direction) {
    // Validation de la réponse actuelle
    const selectedAnswer =
currentQuestion.querySelector('input:checked');
    if (!selectedAnswer && direction > 0) {
        alert('Veuillez sélectionner une réponse');
        return;
    }

    // Masquer question actuelle, afficher suivante
    currentQuestion.style.display = 'none';
    currentQuestionIndex += direction;
    nextQuestion.style.display = 'block';

    // Mettre à jour barre de progression
    const progressPercentage = ((currentQuestionIndex + 1) /
totalQuestions) * 100;
    document.getElementById('progressBar').style.width =

```



```
progressPercentage + '%';  
}
```

3. Ajout dynamique de questions (create-quiz.php)

```
javascript  
function addQuestion() {  
    questionCount++;  
    const questionDiv = document.createElement('div');  
    questionDiv.className = 'question-item';  
    questionDiv.innerHTML = `  
        // Template HTML de la question  
    `;  
    container.appendChild(questionDiv);  
}
```

Avantage : Pas de rechargement de page, interface fluide

4. Filtrage en temps réel (student-dashboard.php)

```
javascript  
function filterQuizzes(subjectId) {  
    const quizCards = document.querySelectorAll('.quiz-card');  
    quizCards.forEach(card => {  
        if (subjectId === 'all' || card.dataset.subject ===  
subjectId) {  
            card.style.display = 'block';  
        } else {  
            card.style.display = 'none';  
        }  
    });  
}
```

RESPONSIVE DESIGN

Media Queries

CSS

```
@media (max-width: 768px) {  
  .quiz-grid {  
    grid-template-columns: 1fr;  
  }  
  .form-row {  
    flex-direction: column;  
  }  
}
```

Adaptations : Grilles en colonne unique, navigation verticale sur mobile

OPTIMISATIONS TECHNIQUES

1. Transitions fluides

CSS

```
transition: all 0.3s ease;
```

Application : Sur tous les éléments interactifs

2. Transform au lieu de propriétés layout

CSS

```
transform: translateY(-10px); /* Plus performant que top: -10px  
*/
```

Avantage : Animations GPU-accélérées

3. Variables CSS pour cohérence

CSS

```
:root {  
  --primary-gradient: linear-gradient(45deg, #667eea, #764ba2);  
  --success-gradient: linear-gradient(45deg, #28a745, #20c997);  
}
```

RÉSULTAT FINAL

- ✓ **Interface moderne** avec animations fluides
- ✓ **Feedback visuel** immédiat sur toutes les interactions
- ✓ **Navigation intuitive** sans rechargement de page
- ✓ **Design cohérent** sur toutes les pages
- ✓ **Performance optimisée** avec CSS3 et JavaScript vanilla