

**RAPPORT TECHNIQUE DU PROJET : Mon Miam Miam**

**Version : V1.0.2**

**Propriétaire :** **ZeDuc@Space** - x - **DNx**



Réalisé par : Groupe 02

Ucac-icam

**ZEDUC\_WEB**

# **MEMBRES DU GROUPE**

* **KENGNE FOKO Aurelle Lydienne (Chef projet) ;**
* **AZEMKOUO TEZANOU Tony ;**
* **MAWAMBA TOWA Maëva ;**
* **NDTOUNGOU NZAMBI Willem Alex ;**
* **TSAPI TIOMELA Elie Junior ;**
* **YACOUB BASSI SOUNOUMOUGOU.**

**SOMMAIRE**

[**MEMBRES DU GROUPE** 2](#_Toc180732560)

[**SOMMAIRE** 3](#_Toc180732561)

[**INTRODUCTION** 4](#_Toc180732562)

[**I-** **OBJECTIFS DU PROJET** 5](#_Toc180732563)

[**1.** **Gestion des utilisateurs** 5](#_Toc180732564)

[**2.** **Gestion des commandes** 5](#_Toc180732565)

[**3.** **Gestion des promotions et offres spéciales** 5](#_Toc180732566)

[**4.** **Interactions dynamiques** 6](#_Toc180732567)

[**5.** **Gestion des événements et jeux** 6](#_Toc180732568)

[**6.** **Tableau de bord pour l’administration** 6](#_Toc180732569)

[**7.** **Sécurité et confidentialité** 6](#_Toc180732570)

[**II-** **ARCHITECTURE DU SYSTEME** 7](#_Toc180732571)

[**III-** **BASE DE DONNEES** 8](#_Toc180732572)

[**IV-** **INTERFACE UTILISATEUR** 9](#_Toc180732573)

[**1.** **Page de connexion** 10](#_Toc180732574)

[**2.** **Tableau de bord des commandes** 10](#_Toc180732575)

[**3.** **Page d'accueil** 10](#_Toc180732576)

[**V-** **FONCTIONNALITES** 11](#_Toc180732577)

[**1.** **Gestion des utilisateurs** 11](#_Toc180732578)

[**2.** **Gestion des commandes** 12](#_Toc180732579)

[**3.** **Promotions et jeux** 12](#_Toc180732580)

[**4.** **Gestion des sessions** 13](#_Toc180732581)

[**VI-** **SECURITE** 13](#_Toc180732582)

[**5.** **Méthodes d'authentification des utilisateurs** 13](#_Toc180732583)

[**6.** **Gestion des autorisations selon les rôles** 14](#_Toc180732584)

[**7.** **Protection contre les failles** 14](#_Toc180732585)

[**VII-** **TESTS** 14](#_Toc180732586)

[**Conclusion** 17](#_Toc180732587)

[**REFERENCES** 18](#_Toc180732588)

# **INTRODUCTION**

La plupart des restaurants modernes utilisent des solutions numériques pour simplifier et automatiser les tâches quotidiennes. C'est dans cette optique que **Zeduc-sp@ce** a fait appel à notre équipe de développeurs pour créer une application web qui servira d'interface entre le restaurant et les étudiants de l'Institut UCAC-ICAM, leurs principaux clients. Le projet **Zeduc\_web** vise à développer une plateforme web intuitive permettant de gérer efficacement les commandes et les utilisateurs. Construite avec le framework **Laravel** pour le backend et **MySQL** comme base de données, cette application permet aux étudiants de s'inscrire, se connecter et gérer leurs sessions en toute sécurité. Elle propose également une interface réactive et évolutive où ils peuvent consulter les repas offerts, profiter des promotions, et participer à des jeux organisés par le restaurant.

1. **OBJECTIFS DU PROJET**

L’application Zeduc\_web a pour objectif principal de faciliter l’interaction entre le restaurant et ses clients réguliers, les étudiants de l’Institut Ucac-Icam. Pour atteindre cet objectif, plusieurs fonctionnalités clés ont été définies, visant à automatiser :

1. **Gestion des utilisateurs**

* **Inscription et connexion des utilisateurs :** l’application permet aux étudiants de créer un compte personnel, de se connecter, et de gérer leurs informations de profil. Le processus d’inscription inclut la validation des informations, comme l’adresse mail et des mots de passe sécurisés.
* **Rôles d’utilisateur :** deux types principaux sont définis (personnel du restaurant) et les utilisateurs (étudiants). Les administrateurs ont des privilèges supplémentaires pour gérer les commandes, les promotions, les employés et autres configurations.

1. **Gestion des commandes**

* **Création de commandes :** les étudiants peuvent parcourir le menu, sélectionner des plats ou des boissons, et passer des commandes directement via l’application. Un système de panier permet de gérer plusieurs commandes avent la finalisation de la commande.
* **Historique des commandes** : Un historique est disponible pour permettre aux étudiants de visualiser leurs anciennes commandes et d'en réitérer si nécessaire.
* **Modification ou annulation de commandes** : Les étudiants peuvent modifier ou annuler une commande tant que celle-ci n’a pas été confirmée par le restaurant.

1. **Gestion des promotions et offres spéciales**

* **Affichage des promotions :** Le restaurant peut publier des promotions ou offres spéciales qui apparaissent sur la page d'accueil des étudiants. Cela permet de promouvoir des repas à prix réduit ou des événements particuliers.
* **Gestion des coupons :** Les étudiants peuvent utiliser des points de fidélité obtenu lorsqu’ils commandent des plats et parraine leurs camarades pour bénéficier des réductions sur certaines commandes.

1. **Interactions dynamiques**

* **Filtres de recherche** : Les utilisateurs peuvent filtrer le menu selon plusieurs critères (type de repas, prix, disponibilité) pour trouver facilement les plats qui correspondent à leurs préférences.

1. **Gestion des événements et jeux**

* **Participation à des jeux ou concours** : Le restaurant peut organiser des jeux ou concours pour les étudiants, et ceux-ci peuvent y participer directement via l'application. Cela peut inclure des tirages au sort, des quiz, ou d'autres formes de compétition.
* **Récompenses et notifications** : Les étudiants peuvent recevoir des notifications sur les jeux en cours, les récompenses disponibles ou les résultats des concours auxquels ils ont participé.

1. **Tableau de bord pour l’administration**

* **Suivi des commandes** : Les administrateurs peuvent voir l’ensemble des commandes passées, leur statut, et gérer leur traitement via un tableau de bord centralisé.
* **Gestion des menus** : Le personnel du restaurant peut modifier le menu, ajouter de nouveaux plats ou en retirer, et ajuster les prix selon les besoins.
* **Gestion des promotions et événements** : Le restaurant peut créer et modifier des promotions ou des événements à partir d'une interface dédiée, tout en surveillant leur impact sur les ventes.

1. **Sécurité et confidentialité**

* **Authentification sécurisée** : Le système garantit une authentification via un hachage des mots de passe pour protéger les informations des utilisateurs.
* **Protection des données personnelles** : L'application respecte les normes de sécurité pour s'assurer que les données des utilisateurs (nom, email, historique de commandes) sont stockées de manière sécurisée.

1. **ARCHITECTURE DU SYSTEME**

L'application **Zeduc\_web** suit une architecture **MVC (Model-View-Controller)**, un modèle largement adopté par le framework **Laravel**. Ce modèle sépare les responsabilités en trois couches distinctes :

* **Modèle (Model)** : Gère la logique des données et les interactions avec la base de données MySQL. Toutes les opérations liées aux commandes, utilisateurs, promotions, etc., sont centralisées dans cette couche.
* **Vue (View)** : Représente la partie visible de l'application pour les utilisateurs. Les pages sont construites en HTML, CSS, Boostrap et JavaScript, et affichent les informations fournies par le modèle.
* **Contrôleur (Controller)** : Sert d'intermédiaire entre la vue et le modèle. Il traite les requêtes utilisateur, appelle les modèles appropriés, et renvoie les vues correspondantes.
* **Schéma d'architecture général**

L'architecture de **Zeduc-sp@ce** se compose de trois couches principales :

* **Frontend (interface utilisateur)** : Utilise HTML, CSS, Bootstrap et **JavaScript** pour créer une interface interactive et réactive. **Bootstrap** est utilisé pour le design et la mise en page, garantissant une interface agréable et responsive sur différents appareils.
* **Backend** : Le backend, géré par **Laravel**, traite la logique métier et les interactions avec la base de données. Il gère les fonctionnalités comme l'inscription, la gestion des commandes et les autorisations.
* **Base de données** : La base de données **MySQL** stocke toutes les informations relatives aux utilisateurs, aux commandes, aux promotions, et aux autres données cruciales de l'application. **AJAX** est utilisé pour des interactions en temps réel, sans rechargement complet des pages, permettant des actions comme la mise à jour des commandes ou l'affichage des promotions de manière fluide.

1. **BASE DE DONNEES**

La base de données de l'application **Zeduc-\_web** est structurée pour faciliter la gestion des utilisateurs, des commandes, des promotions, et des sessions. Elle repose sur **MySQL** et suit un schéma relationnel bien défini avec des tables interconnectées.

* **Modèle de base de données**

Les tables principales de la base de données incluent :

* **Utilisateurs** (users) :
  + Stocke les informations des étudiants et des administrateurs (nom, email, rôle, mot de passe).
  + Champs clés : id, name, email, password, role, created\_at, updated\_at.
* **Commandes** (orders) :
  + Enregistre les commandes passées par les étudiants, associant chaque commande à un utilisateur et des détails comme les plats commandés.
  + Champs clés : id, user\_id (référence à la table users), status (en cours, terminée), total\_price, created\_at, updated\_at.
* **Sessions** (sessions) :
  + Gère les sessions utilisateur pour la connexion et la gestion des sessions actives.
  + Champs clés : id, user\_id, session\_token, last\_activity.
* **Promotions** (promotions) :
  + Contient les informations sur les promotions et offres spéciales publiées par le restaurant.
  + Champs clés : id, title, description, discount, start\_date, end\_date, created\_at, updated\_at.
* Employé :
  + Contient les informations sur les employés du restaurant ;
  + Champs clés : id\_employe, nom, prenom, poste et email.
* Panier :
  + Qui stocke les informations sur les plats de façon temporaire et dispassait lorsque le client valide ou supprime sa commande ;
  + Champs clés : id\_panier, id\_utilisateur, date\_Ajout.
* **Schéma ERD (Entity-Relationship Diagram)**

Le **Schéma ERD** montre les relations entre les tables :

* La table users est en relation **un-à-plusieurs** avec la table orders : un utilisateur peut passer plusieurs commandes.
* La table orders est liée à la table promotions via un lien optionnel, pour lier une commande à une promotion spécifique si elle est appliquée.
* La table sessions est liée à users par une relation **un-à-un** pour gérer les sessions actives de chaque utilisateur.

Cette structure relationnelle permet de garantir une gestion cohérente des données tout en assurant la scalabilité de l’application.

1. **INTERFACE UTILISATEUR**

Les maquettes offrent un aperçu des principales interfaces de l'application de **Zeduc-sp@ce**, permettant de visualiser l'expérience utilisateur et la navigation avant le développement complet. Voici les interfaces principales :

* 1. **Page de connexion**
* **Formulaire de connexion** : Champs pour l'email et le mot de passe, avec un bouton "Se connecter".
* **Lien de récupération de mot de passe** : Accessible en cas d'oubli du mot de passe.
* **Bouton d'inscription** : Redirige vers une page de création de compte.

## **Tableau de bord des commandes**

* **Liste des commandes** : Affiche toutes les commandes passées avec leur statut (en préparation, terminée, annulée).
* **Boutons d’action** : Modifier ou annuler une commande en un clic.
* **Filtre des commandes** : Filtre par statut ou par date.
  1. **Page d'accueil**
* **Menu des plats** : Liste des plats disponibles avec un aperçu rapide (nom, prix, description).
* **Promotions** : Section dédiée aux offres spéciales en cours.
* **Panier** : Icône de panier affichant le nombre d'articles ajoutés et un lien vers la finalisation de la commande.
* **Jeux**: affiche les jeux disponible pour la journée.

**Description de l'interface utilisateur**

L'interface utilisateur est conçue pour être simple et intuitive, facilitant la navigation des utilisateurs dans l'application. Elle se base sur une mise en page fluide et des éléments interactifs bien positionnés.

**Navigation entre les différentes pages**

* **Menu de navigation** : Situé en haut de chaque page, il permet d'accéder rapidement à la page d'accueil, aux commandes, et aux promotions. Il contient également des boutons pour la connexion et la déconnexion.
* **Liens dynamiques** : Les liens vers des sections comme "Promotions" ou "Mon panier" redirigent vers les pages correspondantes sans nécessiter de rechargement complet.

**Présentation des éléments interactifs**

* **Boutons d'action** : Pour la validation des commandes, l'application de promotions, ou la gestion des comptes.
* **Formulaires** : Utilisés pour l'inscription, la connexion, et la validation de commande.
* **Listes déroulantes** : Permettent aux utilisateurs de filtrer les plats par catégorie (plats principaux, boissons, desserts) ou de sélectionner des promotions.

**Composants dynamiques**

L'application **Zeduc-sp@ce** utilise des composants dynamiques pour améliorer l'expérience utilisateur et réduire les temps de chargement :

* **JavaScript** : Gère les interactions sans rechargement de page, notamment pour les filtres de recherche, les mises à jour de panier, et la sélection des catégories dans le menu.
* **AJAX** : Utilisé pour soumettre des formulaires, mettre à jour les commandes, ou appliquer des filtres sans recharger la page. Par exemple, lorsque l'utilisateur sélectionne un plat ou applique une promotion, le panier est mis à jour dynamiquement.

Ces composants dynamiques garantissent une navigation fluide et une interactivité élevée, améliorant ainsi l'engagement des utilisateurs avec l'application.

1. **FONCTIONNALITES**
   1. **Gestion des utilisateurs**

La gestion des utilisateurs comprend les processus d'inscription, de connexion et de déconnexion. Elle repose sur l'implémentation des contrôleurs d'authentification et des routes associés.

* **Inscription et connexion** : Utilise le RegisteredUserController pour gérer l'inscription (register) et le AuthenticatedSessionController pour gérer la connexion (login) et la déconnexion des utilisateurs. Ces contrôleurs interagissent avec la base de données pour créer des utilisateurs ou vérifier leurs identifiants.
* **Récupération de mot de passe** : Nous avons configuré un système de réinitialisation de mot de passe. Ce processus permet aux utilisateurs de recevoir un lien de réinitialisation par email.
* **Gestion des rôles** : Les rôles sont gérés dans la base de données via un champ role (admin, utilisateur, employé, gérant), permettant d'attribuer des autorisations différenciées. Dans web.php, certaines routes sont protégées par des middlewares, pour limiter les accès aux seuls administrateurs ou aux utilisateurs connectés.
  1. **Gestion des commandes**

Les commandes permettent aux étudiants de créer et de suivre leurs commandes, avec une interface intuitive.

* **Création, modification et suppression de commandes** : À travers le GestionMenuController, les étudiants peuvent créer une commande en sélectionnant des plats du menu. Les routes définies dans web.php permettent la modification et la suppression de commandes, avec une vérification des autorisations pour limiter ces actions aux propriétaires des commandes.
* **Suivi des commandes** : L’interface de gestion des commandes permet aux étudiants de visualiser l’état de leur commande (en préparation, livrée). Cette fonctionnalité est rendue possible par un champ de statut dans la base de données, et les étudiants peuvent accéder aux informations de suivi de leurs commandes à partir de la page gestionmenu.
  1. **Promotions et jeux**

Cette fonctionnalité enrichit l’expérience utilisateur par des offres promotionnelles et des événements interactifs.

* **Affichage des offres promotionnelles** : Les promotions sont affichées sur les pages d’accueil et de menu, et sont configurables par les administrateurs dans la base de données. Les étudiants sont ainsi informés des offres en cours via des bannières ou des notifications visuelles.
* **Interaction avec les jeux et concours** : La fonctionnalité de parrainage est mise en place via un système de génération de liens uniques pour chaque utilisateur. En cliquant sur un bouton « Partager mon lien », un code de parrainage et un QR code sont générés et partagés. Cette interaction est gérée par un script JavaScript qui s’assure que chaque utilisateur dispose d’un code unique.
  1. **Gestion des sessions**

Les sessions utilisateur sont gérées principalement via la base de données pour offrir une persistance des données et améliorer la sécurité.

* **Sessions basées sur la base de données** : Les sessions sont enregistrées dans la base de données, offrant une sécurité accrue et permettant une gestion centralisée des connexions utilisateurs. Cela permet également de suivre les connexions des utilisateurs et de les déconnecter à distance si nécessaire, en particulier dans le cas de rôles administratifs.
* **Cookies pour une meilleure expérience utilisateur** : Les cookies sont utilisés pour maintenir les préférences de l'utilisateur et pour des éléments persistants comme le panier des commandes ou les paramètres de recherche.

Ces fonctionnalités principales sont réalisées en combinant des composants Laravel (routes, contrôleurs, middlewares) avec une interface front-end en HTML, CSS, et JavaScript, intégrée via Bootstrap pour un design réactif et accessible. Les composants de Laravel sont utilisés de manière à séparer la logique métier et à assurer une navigation sécurisée.

1. **SECURITE**

L'application de **Zeduc-sp@ce** repose sur des méthodes robustes pour gérer l'authentification et l'autorisation des utilisateurs afin de garantir la sécurité et la gestion des accès.

* 1. **Méthodes d'authentification des utilisateurs**

L'application utilise le système d'authentification intégré de **Laravel Auth** pour gérer l'inscription et la connexion des utilisateurs. Les principales fonctionnalités incluent :

* **Inscription** : Les utilisateurs peuvent créer un compte avec leur nom, email, et un mot de passe sécurisé.
* **Connexion** : Les utilisateurs peuvent se connecter en fournissant leur email et mot de passe.
* **Réinitialisation du mot de passe** : Un système de récupération de mot de passe via email est mis en place pour les utilisateurs ayant oublié leur mot de passe.
  1. **Gestion des autorisations selon les rôles**

L'application met en place une gestion des rôles pour accorder des autorisations spécifiques :

* **Administrateur (admin)** : Accès complet à l'application, y compris la gestion des utilisateurs, des commandes et des promotions.
* **Utilisateur (client)** : Accès limité à l'interface pour consulter et passer des commandes, et visualiser les promotions.

Les rôles sont gérés via des **policies** et **middleware** dans Laravel pour restreindre l'accès à certaines sections de l'application en fonction des privilèges.

* 1. **Protection contre les failles**

**CSRF (Cross-Site Request Forgery)**

Laravel protège automatiquement contre les attaques CSRF en incluant un **jeton CSRF** dans chaque formulaire généré. Ce jeton est vérifié à chaque soumission pour s'assurer que la requête provient bien de l'utilisateur légitime.

**Validation des entrées utilisateurs**

Des mesures strictes sont mises en place pour prévenir les failles de sécurité :

* **Validation XSS (Cross-Site Scripting)** : Les données fournies par les utilisateurs sont filtrées et échappées pour éviter l'injection de scripts malveillants.
* **SQL Injection** : Laravel utilise des **requêtes préparées** et l'ORM Eloquent, qui gèrent automatiquement l'échappement des entrées utilisateurs, protégeant ainsi contre les injections SQL.

1. **TESTS**

Les **tests unitaires** sont essentiels pour garantir que chaque composant de l'application fonctionne correctement de manière isolée. Dans **Zeduc\_web**, les tests unitaires sont écrits avec l’outil de test intégré de **Laravel** (PHPUnit). Ils couvrent les fonctionnalités suivantes :

* **Authentification** : Vérification que les utilisateurs peuvent s’inscrire, se connecter et réinitialiser leurs mots de passe.
* **Gestion des commandes** : Tests des fonctions de création, modification et suppression des commandes.
* **Promotions** : Validation de l’application correcte des promotions à une commande.
  + **Tests d'intégration**

Les **tests d’intégration** s'assurent que les différents composants de l'application fonctionnent bien ensemble. Ils incluent des scénarios tels que :

* **Connexion d’un utilisateur** puis création d'une commande.
* **Affichage des promotions** et ajout d’une promotion à une commande. Ces tests vérifient la bonne interaction entre le backend (Laravel), la base de données (MySQL), et le frontend.
  + **Scénarios de test utilisateur**

Des **tests manuels** sont réalisés pour vérifier l’interface utilisateur et les principales fonctionnalités :

* **Inscription et connexion** : Test du processus depuis la création de compte jusqu'à l’accès au tableau de bord.
* **Navigation entre les pages** : Vérification de la fluidité de la navigation entre les différentes sections (menu, panier, commandes).
* **Gestion des commandes** : Test du processus de commande, du panier à la confirmation finale.
  + **Déploiement**

**Environnement de production**

Pour le déploiement en environnement de production, l'application est hébergée sur **Vercel**, un service de déploiement cloud optimisé pour les applications web.

* **Hébergement** : Vercel gère le déploiement continu de l’application, garantissant que chaque nouvelle version est mise en production sans interruption.
* **Configuration de l’environnement** : Des variables d’environnement (comme les identifiants de la base de données ou les clés API) sont configurées directement dans l'interface Vercel pour sécuriser l’application.
  + **Processus de déploiement**
* **Utilisation de Git** : Le code source de l’application est géré avec **Git**, et les branches principales (ex. main) sont automatiquement déployées sur Vercel grâce à une intégration continue (CI/CD). Toute modification apportée sur la branche principale déclenche un déploiement automatique.
* **Procédures de sauvegarde et restauration de la base de données** : Des sauvegardes régulières de la base de données sont effectuées pour garantir la récupération en cas de problème. Des outils comme **mysqldump** peuvent être utilisés pour exporter et restaurer la base de données si nécessaire.

Ce processus de déploiement assure que l’application reste toujours à jour et que les utilisateurs bénéficient des dernières fonctionnalités sans interruption.

Haut du formulaire

Bas du formulaire

# **Conclusion**

Le projet **Zeduc-sp@ce** vise à offrir une solution numérique moderne pour améliorer l'expérience des étudiants de l'Institut UCAC-ICAM lors de leurs interactions avec le restaurant. Grâce à une interface intuitive et un backend robuste, l'application permet une gestion simplifiée des commandes et des utilisateurs, tout en offrant des promotions et des services adaptés à la clientèle cible. Développée avec **Laravel** et intégrant des technologies sécurisées comme **MySQL** pour la base de données et **AJAX** pour des interactions dynamiques, l'application garantit à la fois performance et évolutivité.

Avec un processus rigoureux de tests, de sécurité renforcée et un déploiement continu sur **Vercel**, Zeduc-sp@ce est conçue pour répondre aux besoins croissants d’un restaurant universitaire tout en offrant une expérience utilisateur fluide. Ce projet, en plus de contribuer à l'efficacité opérationnelle, place l'établissement en première ligne des innovations numériques dans le domaine de la restauration.

# **REFERENCES**

* + Maquette Figma de l’application zeduc\_web

[Figma](https://www.figma.com/design/fjLSZqoXJtJsOJp7DhUPBQ/ZEDUC-SP%40CE-2?node-id=0-1&node-type=canvas&t=GP9jIYCNVbYeswHs-0)

* + Lien du dépôt git

[Foko860/ZEDUC\_WEB (github.com)](https://github.com/Foko860/ZEDUC_WEB)

* + Lien du JIRA

<https://2028-team-d1jsrg3v.atlassian.net/jira/software/projects/ZW/boards/3/reports/burnup>