

# Rapport de projet - CheckMate Nasseraldin Ramie

---

## 1. Technologies utilisées (et justification)

---

- **Go 1.24 + Gin + Gorm/SQLite** : pile légère, un binaire + un fichier DB unique ( `data.db` ), maintien facile du nombre de fichiers côté serveur (routes/main/models). Sessions par cookie `HttpOnly`.
- **Templates HTML + Tailwind CSS compilé** : CSS déjà générée dans `static/css` .
- **JavaScript vanilla** : bascule langue (fr/en) et thème jour/nuit (avec tailwind) sans rechargement, appels `fetch JSON` pour la liste/suppression/creation de dépenses.
- **Dépassement autorisé** : la compilation Tailwind produit quelques fichiers CSS/JS supplémentaires dans `static/` . Le cœur serveur reste <10 fichiers non binaires ; côté client les templates+JS principaux restent compacts.

## 2. Manuel d'utilisation (navigation rapide)

---

- Lancer le serveur : `go mod download && go run .` puis ouvrir `http://localhost:8080` .
- Comptes : `demo@example.com` / `demo` (utilisateur), `admin@example.com` / `admin` (admin).
- Pages :
  - `/` : accueil + stats synthétiques (Pour l'admin, c'est le cumul de toutes les dépenses enregistrés).
  - `/login` et `/register` : auth.
  - `/expenses` : liste personnelle (ou globale pour admin) avec suppression.
  - `/expenses/create` : ajout d'une dépense (admin peut choisir l'utilisateur).
  - `/admin` : tableau de bord admin (stats globales, utilisateurs, dernières dépenses).
- Boutons persistants (toutes pages) :
  - **Langue** : globe (fr/en, stockage `localStorage`).
  - **Thème** : lune/soleil (clair/sombre, stockage `localStorage`).
- Requêtes BD :
  - Lecture (SELECT) via `/get_expenses_by_category` (liste avec jointure User).
  - Écriture (INSERT) via `/expense/create_expense` , suppression (DELETE) via `/expenses/delete/:id` .

## 3. Détails techniques

---

- **Préférences sans rechargeement** : `static/js/preferences.js` lit `localStorage` dès le `<head>` , ajuste `document.documentElement` (lang + classe `dark`) avant le rendu ; les boutons mettent à jour les attributs `data-i18n` et le DOM en direct.
- **Seed et comptes par défaut** : `models.go` appelle `initDatabase` qui auto-migre, crée un utilisateur démo et un admin si absents, et injecte deux dépenses de départ.
- **Gestion des rôles** : middleware `requireAuth` et `requireAdmin` dans `routes.go` ; les requêtes JSON filtrent par utilisateur sauf admin ( `/get_expenses_by_category` ).
- **Formulaires et API JSON** : routes acceptent à la fois `application/json` et POST formulaire ; la vue est rechargée ou renvoie JSON selon `Accept / Content-Type` .

- **Structure minimale** : serveur limité à `main.go`, `routes.go`, `models.go`. Côté client : templates dans `templates/`, JS dans `static/js/`, CSS précompilée dans `static/css/`.
- **Modèles et base** : 2 tables Gorm/SQLite simples. `User` (ID, Name, Email unique, PasswordHash, IsAdmin, relation `Expenses`) et `Expense` (ID, Title, Amount, Currency, DateAchat, Category, UserID + relation `User`). Mots de passe hashés (bcrypt), contraintes `OnUpdate:CASCADE`, `OnDelete:RESTRICT` pour éviter les effacements involontaires. `initDatabase` auto-migre, crée un compte démo + admin si absents et injecte deux dépenses de seed si nécessaire.

## 4. Validation rapide (Firefox)

---

- Connexion/déconnexion + création d'un compte.
- Créer une dépense, vérifier son apparition dans `/expenses`, tester suppression.
- En admin : accéder à `/admin`, vérifier stats et dernières dépenses.
- Tester bascule fr/en et clair/sombre sans recharge (persistance après refresh).
- Vérifier les réponses JSON (Accept: application/json) pour création/liste.

## 5. Sécurité et limites

---

- Hash bcrypt des mots de passe ; sessions HttpOnly.
- CSRF non implémenté (à ajouter si déploiement public).
- Validation basique (types/requis) ; pas de vérification email.

## 6. Pistes d'amélioration

---

- CSRF + rate limiting + logging structuré.
- Pagination/recherche sur `/expenses` ; filtres par période.
- Exports CSV/JSON, budgets par catégorie, notifications.
- Tests d'intégration Gin/SQLite.

# Admin Panel /admin

## Admin space

Manage users and monitor global spending.

**USERS**  
**4**  
Registered accounts

**EXPENSES**  
**6**  
Total records

Users		
NAME	EMAIL	EXPENSES
Administrateur ADMIN	admin@example.com	0
Utilisateur Démo	demo@example.com	5
grrr paw	aaa@gmail.com	0
kngjenudnn	rnioug@gmail.com	1

**Latest expenses**

hhh kngjenudnn - 2025-11-07	8.00 EUR
--------------------------------	----------

Total records

Users		
NAME	EMAIL	EXPENSES
Administrateur ADMIN	admin@example.com	0
Utilisateur Démo	demo@example.com	5
grrr paw	aaa@gmail.com	0
kngjenudnn	rnioug@gmail.com	1

**Latest expenses**

hhh kngjenudnn - 2025-11-07	8.00 EUR
Mac Utilisateur Démo - 2025-11-07	1199.00 EUR
Urssaf Utilisateur Démo - 2025-10-08	305.00 EUR
Navigo Utilisateur Démo - 2025-10-08	92.50 EUR
Pizza Utilisateur Démo - 2025-10-01	24.50 EUR
Steam Utilisateur Démo - 2025-09-08	40.00 EUR

## Créer une dépense (user) /expenses/create

The screenshot shows the 'Create an expense' page of the CheckMate application. At the top, there's a navigation bar with links for 'Home', 'Add an expense', 'Expenses', and a user profile 'Utilisateur Démo'. There are also 'Sign out' and 'Switch to French' buttons, along with a small user icon.

The main form has the following fields:

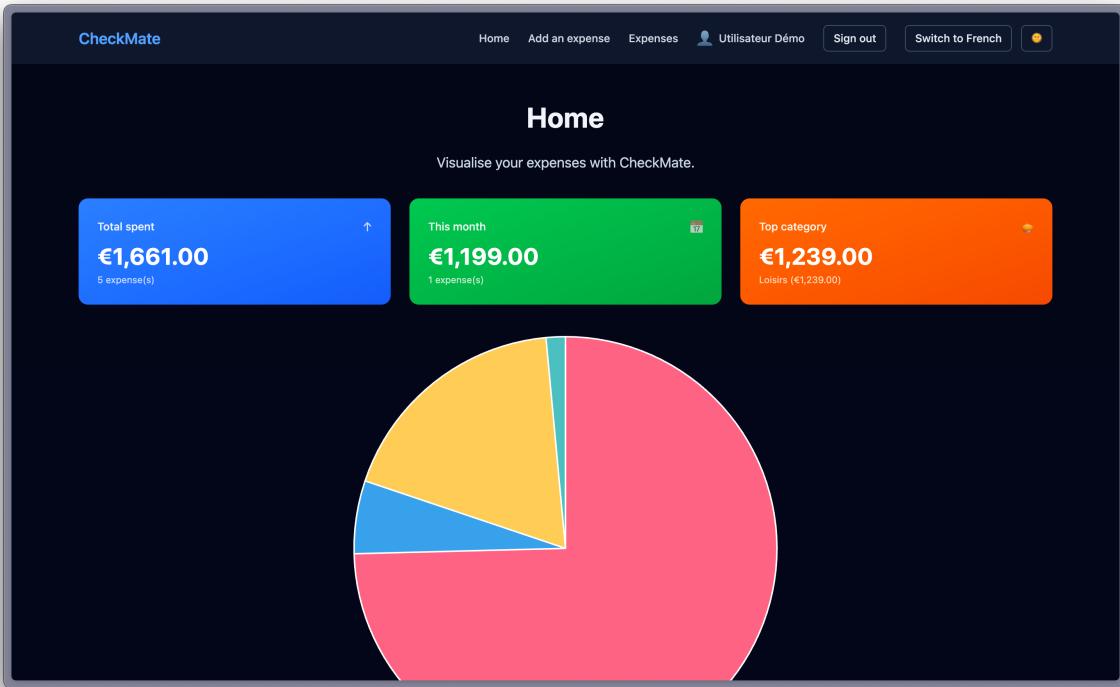
- Title:** A text input field with placeholder text 'e.g. Pizza, Uber...'.
- User:** A dropdown menu showing '✓ Utilisateur Démo'.
- Amount:** An input field containing '0.00'.
- Currency:** A dropdown menu showing 'EUR'.
- Category:** A dropdown menu showing 'Housing'.
- Purchase date:** A date input field showing '20/11/2025'.

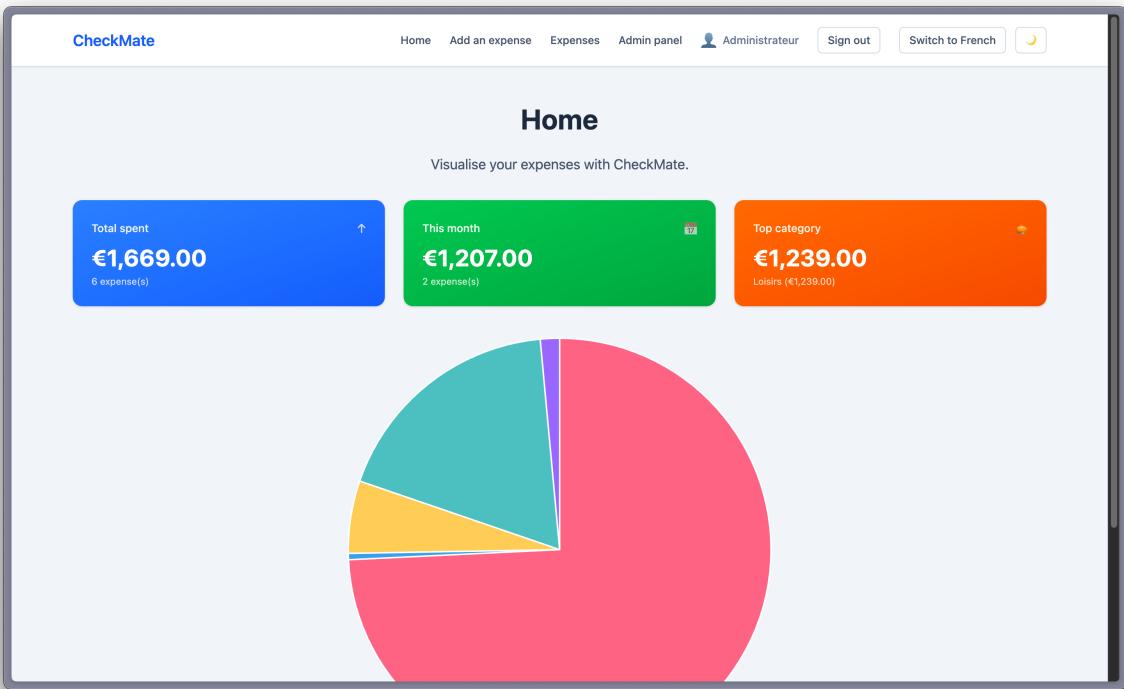
At the bottom right of the form are two buttons: 'Reset' and 'Add', with 'Add' being highlighted in blue.

## Footer



## Accueil (fr/en, dark/light)





### Liste des dépenses user /expenses

**CheckMate**

Home Add an expense Expenses Utilisateur Démô Sign out Switch to French ☀️

#### Expenses list

Back to home

DESCRIPTION	USER	CATEGORY	DATE	AMOUNT	ACTIONS
Pizza	Utilisateur Démô	Nourriture	2025-10-01	24.50 EUR	trash
Navigo	Utilisateur Démô	Transport	2025-10-08	92.50 EUR	trash
Urssaf	Utilisateur Démô	Impots	2025-10-08	305.00 EUR	trash
Steam	Utilisateur Démô	Loisirs	2025-09-08	40.00 EUR	trash
Mac	Utilisateur Démô	Loisirs	2025-11-07	1199.00 EUR	trash

## Login / Register (/login, /register)

