



---

# Rapport technique sur invoicesGenerator

---

**Écrit par:**

BENAISSA Meriem 21418006

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du projet</b>	<b>2</b>
<b>3</b>	<b>Architecture de l'application</b>	<b>2</b>
<b>4</b>	<b>Modélisation de la base de données</b>	<b>2</b>
4.1	Product . . . . .	2
4.2	Invoice . . . . .	3
4.3	InvoiceItem . . . . .	3
<b>5</b>	<b>Fonctionnalités principales</b>	<b>3</b>
5.1	Gestion des produits . . . . .	3
5.2	Gestion des factures . . . . .	3
<b>6</b>	<b>Génération PDF avec ReportLab</b>	<b>4</b>
<b>7</b>	<b>Interface utilisateur</b>	<b>4</b>
<b>8</b>	<b>Technologies utilisées</b>	<b>4</b>
<b>9</b>	<b>Déploiement local</b>	<b>4</b>

---

# 1 Introduction

Ce projet a pour objectif de concevoir une application web avec **Django** permettant de gérer une base de produits et de générer dynamiquement des **factures PDF**. L'application offre une interface simple et efficace destinée à des utilisateurs souhaitant effectuer une gestion basique de ventes et de factures.

## 2 Présentation du projet

L'application, nommée *InvoiceGenerator*, propose les fonctionnalités suivantes :

- Gestion complète des produits (ajout, édition, suppression)
- Création et affichage de factures
- Export de factures en PDF via ReportLab
- Interface utilisateur responsive avec Bootstrap

## 3 Architecture de l'application

L'application est construite selon l'architecture **MVC (Modèle-Vue-Contrôleur)** de Django :

- **Modèles** : définition des données (Product, Invoice, InvoiceItem)
- **Vues** : traitement des requêtes HTTP et logique métier
- **Templates** : pages HTML affichées à l'utilisateur. Le template utilisé provient d'un support fourni pendant ma formation "*Développeur Python*" chez *Amis de Java*, durant les séances sur le développement d'applications web avec Django. Je l'ai ensuite adapté et modifié pour correspondre aux besoins spécifiques de ce projet, sans me limiter à un simple copier-coller.
- **URLs** : routage entre les vues et les pages web

## 4 Modélisation de la base de données

### 4.1 Product

- id : Identifiant
- name : Nom du produit

- price : Prix unitaire
- expiration\_date : Date d'expiration

## 4.2 Invoice

- id : Identifiant
- date\_created : Date de création
- get\_total() : Calcule le total de la facture

## 4.3 InvoiceItem

- invoice : Référence à une facture
- product : Référence à un produit
- quantity : Quantité commandée
- get\_prix\_total() : Prix total de l'item

# 5 Fonctionnalités principales

## 5.1 Gestion des produits

L'utilisateur peut :

- Ajouter un produit via formulaire
- Modifier un produit existant
- Supprimer un produit
- Rechercher un produit par ID

## 5.2 Gestion des factures

- Création de factures à partir d'une sélection de produits
- Affichage des factures paginées
- Téléchargement des factures en format PDF
- Consultation détaillée des contenus

---

## 6 Génération PDF avec ReportLab

- Titre : Numéro et date de la facture
- Tableau : Produit, prix, quantité, total ligne
- Total : Montant total de la facture
- Mise en page propre et centrée
- Téléchargement via une vue Django avec ‘HttpResponse’

## 7 Interface utilisateur

- Design basé sur Bootstrap + style CSS personnalisé
- Navigation fluide entre dashboard, produits et factures
- Interaction JavaScript pour recherche et génération dynamique

## 8 Technologies utilisées

- Python 3.x et Django
- HTML5 / CSS3 / Bootstrap
- JavaScript
- MySQL
- ReportLab (PDF)

## 9 Déploiement local

Consulter le fichier README.md