

TP2 programmation Web
Site e-commerce en ligne
Fahd et Youssef

- I. Introduction
- II. Cahier des charges
 - 2.1 Présentation de l'application
 - 2.2 Présentation du projet
 - 2.3 Spécifications techniques
 - 2.4 Contraintes & sécurité
- III. Arborescence du site
- IV. Wireframe / organisation des pages
- V. Planche de tendance (concept & inspiration)
- VI. Charte graphique
 - 6.1 Couleurs
 - 6.2 Typographies
 - 6.3 Boutons & composants UI
- VII. Fonctionnalités détaillées (technique)
- VIII. Mockup / aperçu final

- **SOMMAIRE**

- INTRODUCTION

L'objectif du projet TP2 (Programmation Web Avancée) est de développer une **application e-commerce complète**, permettant à un utilisateur de consulter des produits, les ajouter au panier, créer un compte, se connecter, passer une commande et effectuer un paiement en ligne (Stripe ou PayPal). Comme dans l'exemple Bonneaux (page 3 du PDF) , la méthodologie suivie :

1. Analyse des besoins et définition du parcours utilisateur.
2. Conception fonctionnelle : routes, navigation, pages, sécurité.
3. Mise en place du backend (Node.js + Express + API).
4. Développement du frontend (React + Context API).
5. Intégration du système de paiement sécurisé.
6. Tests, validation et corrections.

CAHIER DES CHARGES

➤ Présentation de l'application

Application e-commerce développée par :

Fahd Mammou & Youssef Bssat— TP2 Programmation Web Avancée

- ✓ Le site permet aux utilisateurs de :
- ✓ Consulter des produits
- ✓ Voir la page d'un produitAjouter au panier
- ✓ Se connecter / créer un compte
- ✓ Passer une commande
- ✓ Payer avec Stripe

CAHIER DES CHARGES

➤ Présentation du projet

- ❖ **Objectifs du projet**
 - ❖ Apprendre à développer un front React complet.
 - ❖ Manipuler Context API & gestion d'état globale.
 - ❖ Créer un backend sécurisé avec Node.js / Express.
 - ❖ Intégrer une API de paiement réelle.
 - ❖ Gérer les erreurs, les statuts et les redirections.
- ❖ **Cibles du projet**
 - ❖ Utilisateurs en ligne
 - ❖ Clients voulant acheter des produits
 - ❖ Étudiants et enseignants (correction du TP)
- ❖ **Type d'application**
 - ❖ Site **e-commerce** avec paiement sécurisé.
- ❖ **Pages prévues**
 - ❖ Accueil
 - ❖ Panier
 - ❖ Paiement
 - ❖ Succès / Annulation
 - ❖ Login / Register

CAHIER DES CHARGES

➤ Spécifications techniques

Aspect visuel

Le site doit être :

- Responsive (mobile, tablette, desktop)
- Utiliser une charte graphique cohérente
- Avoir une navigation intuitive

Backend

Technologies utilisées :

- Node.js
- Express.js
- MySQL

Stripe Payment Gateway

Dossiers principaux :

- /config/db.js → connexion SQL
- /controllers/* → logique (auth, ouvrages, paiements)
- /models/* → modèles SQL
- /routes/*
- .env → variables (clé Stripe, DB, JWT, etc.)
- Server.js
- Package.json
- Package-lock.json

CAHIER DES CHARGES

➤ Spécifications techniques

Frontend

Selon l'architecture :

React, ou

HTML/CSS/JS classique avec fetch()

Contenu prévu :

Page d'accueil

Page panier

Page paiement

Page connexion / inscription

Sécurité

Hashage des mots de passe

JWT Tokens

Validation backend

Protection CORS

Systèmes externes

Stripe : création d'intention de paiement

MySQL : stockage des utilisateurs + ouvrages + paiements

CAHIER DES CHARGES

➤ Budget (Simulation pédagogique)

Description	Durée	Coût / jour	Total
Conception + arborescence	2 jours	0\$	0\$
Maquettes XD	2 jours	0\$	0\$
Backend complet	4 jours	0\$	0\$
Frontend complet	4 jours	0\$	0\$
Tests et intégration Stripe	2 jours	0\$	0\$
Total	14 jours	—	0\$ (évaluation académique)

ARBORESCENCE DU SITE

/Frontend → Application React (Vite)

/Backend → API Node.js + Express + Stripe

Architecture du Frontend (React + Vite)

/src

 └── api/

 → appels API backend (axios)

 └── components/

 → Navbar, Footer, Cards,

Modales

 └── context/

 → AuthContext, CartContext

 └── pages/

 → toutes les pages du site

 └── App.jsx

 → routes + providers

 └── main.jsx

 → point d'entrée Vite

 └── assets/

 → images

Architectutre Backend

/Backend

 └── server.js

 └── routes/

 └── auth.routes.js

 └── books.routes.js

 └── payment.routes.js

 └── controllers/

 └── auth.controller.js

 └── books.controller.js

 └── payment.controller.js

 └── config/

 └── stripe.js

 └── data/

 └── books.json (base de données livres)

WIREFRAME

Desktop

Header avec logo + menu

Section principale : affichage des ouvrages

Sidebar (si utilisée)

Footer simple

Tablette

Menu compressé

Contenu central ajusté

Mobile

Menu burger

Blocs verticalisés

Boutons pleine largeur

PLANCHE DE TENDANCE (MOODBOARD)

L'ambiance du projet tourne autour de :

Simplicité

Modernité

Efficacité

Couleurs sobres

UI claire et lisible

Mots-clés :

**Modern / Minimal / Flat Design / Blue
tones / White space**

CHARTE GRAPHIQUE

– Logotype

Le logo utilisé :

Simple

Minimal

Aligné avec le thème de l'application (TP2 – Projet Web)

- Couleurs

#1A73E8 – Bleu primaire

#0D47A1 – Bleu foncé

#FFFFFF – Blanc

#F5F5F5 – Gris clair

#222222 – Noir texte

– Typographies

Titre : Montserrat / 700

Paragraphes : Inter / 400

Boutons : Arial Bold (si demandé)

– Boutons / UI

Bouton principal

Fond bleu (#1A73E8)

Texte blanc

Coins arrondis

Bouton secondaire

Fond blanc

Bordure bleue

Texte bleu

-Liens

Couleur : #1A73E8

Survol : souligné

CHARTE GRAPHIQUE

– Logotype

Le logo utilisé :

Simple

Minimal

Aligné avec le thème de l'application (TP2 – Projet Web)

- Couleurs

#1A73E8 – Bleu primaire

#0D47A1 – Bleu foncé

#FFFFFF – Blanc

#F5F5F5 – Gris clair

#222222 – Noir texte

– Typographies

Titre : Montserrat / 700

Paragraphes : Inter / 400

Boutons : Arial Bold (si demandé)

– Boutons / UI

Bouton principal

Fond bleu (#1A73E8)

Texte blanc

Coins arrondis

Bouton secondaire

Fond blanc

Bordure bleue

Texte bleu

-Liens

Couleur : #1A73E8

Survol : souligné

MAQUETTES GRAPHIQUES

Basées sur le fichier Adobe XD :

Page d'accueil (liste d'ouvrages)

Page login

Page paiement Stripe

Page de connexion

Page d'inscription

Structure :

Header fixe

Sections espacées

Grilles cohérentes

Formulaires alignés