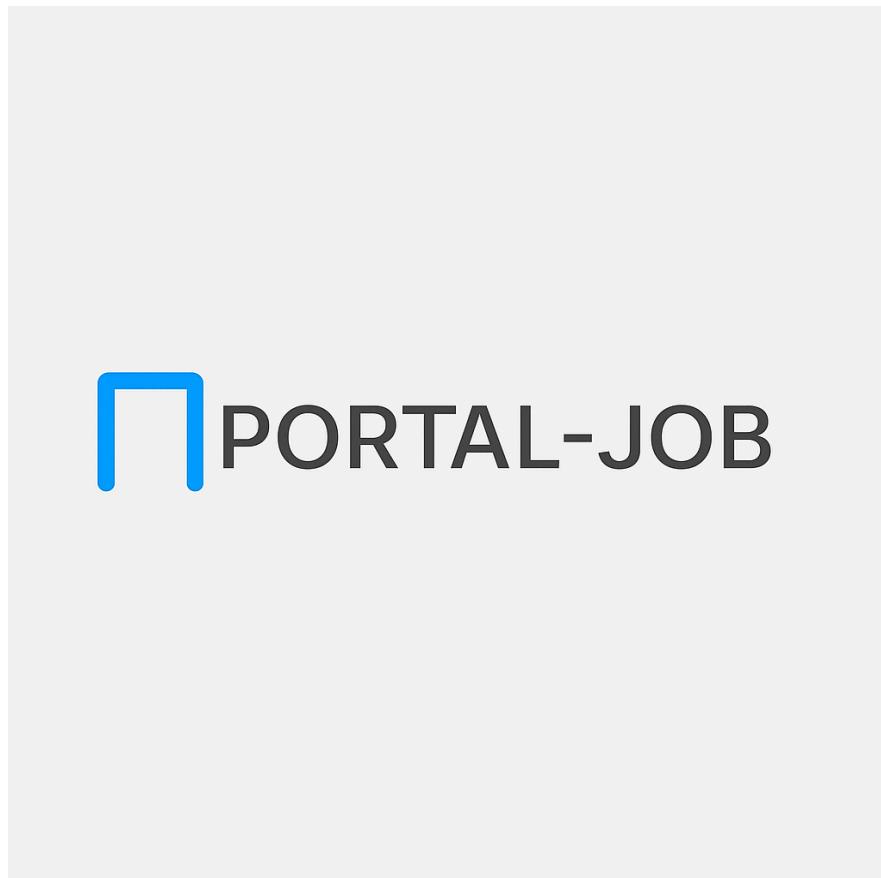


Projet Professionnel

Développeur Web & Web Mobile

REFONTE DU SITE PORTAL_JOB



 PORTAL-JOB



Avant-Propos

Bienvenue dans le projet de refonte du site **Portal_job**

Cette initiative ambitieuse vise à moderniser et améliorer l'expérience en ligne des offres d'emplois.

Mon **objectif** est donc de concevoir un site web qui soit à la fois moderne et digne de l'importance de ce projet, en se basant sur une analyse approfondie des besoins des utilisateurs et des tendances actuelles du marché.

La refonte n'est pas seulement esthétique, elle est **stratégique**, visant à **optimiser** la navigation, la **recherche** d'offres, et l'**efficacité** globale pour les candidats et les recruteurs.

Sommaire du Mémoire

Partie 1 : RNCP 37273 BC01 : Préparer la réalisation d'un site ou d'une application web

1.1 Identification des besoins et des spécifications techniques du projet

1.2 Formalisation du dossier de spécifications et du planning

1.3 Choix des technologies, des outils de collaboration et du mode de travail

1.4 Prise en compte de l'éco-conception, de l'accessibilité et de la sécurité

Partie 2 : RNCP 37273 BC02 : Développer des interfaces Frontend pour un site ou une application web / web mobile

2.1 Découpage de la maquette graphique et mise en place de la structure HTML

2.2 Création des feuilles de style (CSS) et intégration du Responsive Design

2.3 Dynamisation de l'expérience utilisateur et interaction avec le serveur

2.4 Structuration du code et amélioration de l'écriture des fichiers

Partie 3 : RNCP 37273 BC03 : Développer le Backend d'un site ou d'une application web / web mobile

3.1 Conception et création de la base de données (BDD)

3.2 Développement de l'application et de la logique métier côté serveur

3.3 Mise en place des mesures de sécurité de l'application

3.4 Stratégie de tests et processus de maintenance de l'application

Partie 4 : Mise en production et optimisation du projet (Marc ??)

4.1 Choix de l'hébergeur / nom de domaine

4.2 La base de donnée en ligne

4.3 Mise en ligne via un client

4.4 Contrôles performances / SEO

Partie 5 : Annexes

Partie 6 : Remerciements

Partie 1 :

RNCP 37273 BC01 :

**Préparer la réalisation d'un site ou
d'une application web**

L'étape de préparation et de conception est la pierre angulaire de tout projet de développement Full Stack. Elle est non seulement importante pour garantir la faisabilité technique, mais elle pose également les fondations d'un développement sécurisé, performant et conforme aux standards professionnels modernes.

Cette partie du mémoire vise à démontrer la maîtrise de l'intégralité du processus d'ingénierie en amont, faisant du développeur un véritable architecte de solution, capable d'anticiper les défis et d'aligner la vision métier sur la réalité technique.

Le passage de la Vision à la Structure

Le premier défi de ce bloc de compétences consiste à opérer la transformation de l'expression des besoins, souvent exprimée en termes fonctionnels par le client ou le *Product Owner*, en un **plan de développement précis et mesurable**. Cela exige une analyse critique pour **identifier les informations nécessaires** à la réalisation complète du projet : définition du public cible, contraintes d'expérience de navigation, exigences d'accessibilité et de référencement, ainsi que les limites budgétaires et temporelles. C'est à partir de cette analyse rigoureuse que le développeur traduit

l'expression des besoins pour déterminer les **spécifications techniques** détaillées de l'application.

La qualité de cette phase se reflète dans la capacité à **formaliser les spécifications** pour garantir que la solution développée réponde à la problématique initiale. Cette formalisation prend la forme d'un **Dossier de Spécifications Techniques (DST)**, qui ne se contente pas de lister les fonctionnalités, mais qui justifie également les choix fondamentaux d'architecture.

Organisation, Éthique et Pérennité

Au-delà des aspects purement techniques, la préparation englobe une dimension organisationnelle et éthique indispensable dans l'écosystème numérique actuel. Le développeur Full Stack doit **choisir un mode et des outils de travail adaptés** (par exemple, un gestionnaire de version comme Git) pour assurer un suivi fluide de l'avancement et faciliter l'interaction, même dans le cadre d'une équipe polyvalente. La capacité à **délimiter le périmètre de chaque membre** de l'équipe (s'il y a lieu) et à planifier le travail collaboratif est une preuve de leadership technique.

De plus, la phase de préparation est le moment opportun pour intégrer les préoccupations sociétales. Cela inclut l'intégration des **principes d'éco-conception** pour minimiser l'impact environnemental de la solution, l'application stricte des **normes**

d'accessibilité pour garantir l'inclusion des personnes en situation de handicap, et l'établissement des protocoles de sécurité dès l'architecture pour prévenir les vulnérabilités futures. Cette approche globale, où les critères non fonctionnels (sécurité, performance, accessibilité) sont considérés au même titre que les critères fonctionnels, est le gage d'un projet robuste et professionnel.

En conclusion, la validation de ce bloc RNCP 37273 BC01 ne se fait pas par un simple document, mais par la démonstration d'une réflexion stratégique complète qui anticipe l'ensemble du cycle de développement et de maintenance. Cette approche garantit que les solutions proposées sont non seulement techniquement réalisables, mais également pérennes et alignées avec les objectifs métiers de l'organisation.

1.1

Identification des besoins et des spécifications techniques du projet

L'efficacité du développement dépend de la clarté et de la robustesse de sa phase de **spécification**. Pour ce projet, dont l'objectif est de créer une **plateforme de gestion d'offres d'emploi**, l'enjeu majeur était de proposer une interface **simple et ergonomique** afin de faciliter la mise en relation entre recruteurs et chercheurs d'emploi.



Présentation au client interne / Product Owner

🎯 Projet : Plateforme de gestion d'offres d'emploi

1. 📋 Contexte et objectif du projet

Ce projet consiste à développer une **application web** permettant à des **entreprises** de publier des offres d'emploi et à des **candidats** de postuler en ligne. L'objectif est de proposer une interface simple et ergonomique pour faciliter la mise en relation entre recruteurs et chercheurs d'emploi.

Client interne : Directeur de campus (joue le rôle de commanditaire / product owner)

Développeur : Étudiant en BTS – responsable du développement web

Analyse Fonctionnelle et Profils Utilisateurs

L'analyse de la demande initiale, menée avec le Directeur de campus agissant comme *Product Owner*, a permis d'identifier trois profils d'utilisateurs distincts, chacun nécessitant un ensemble de fonctionnalités spécifiques. L'**administrateur** est responsable de la gestion des comptes, de la gestion des offres et de la modération des contenus en gros à la plupart des choses voir tout. Le **recruteur (entreprise)** doit pouvoir créer un profil entreprise et disposer de toutes les fonctions CRUD (Création, Lecture, Modification, Suppression) sur ses offres d'emploi. Enfin, le **candidat** a pour mission de créer son propre profil, de consulter les offres et, surtout, de postuler en ligne, avec la possibilité d'envoyer un CV ou un fichier PDF.

2. Spécifications fonctionnelles principales

Utilisateurs :

- **Administrateur** : gère les comptes, les offres et modère les contenus
- **Recruteur** : créer un profil entreprise, publie/modifie/supprime ses offres
- **Candidat** : crée un profil, consulte les offres, postule avec CV en ligne

Ces rôles ont conduit à la définition des fonctionnalités principales : une **authentification** (Inscription/Connexion) sécurisée pour les trois types d'utilisateurs, la création de **profils distincts** selon le rôle gérer en amont, une **recherche multicritères** performante (poste, lieu,

type de contrat) pour les Candidats , et un **tableau de bord** pour l'Admin qui lui permettra d'avoir la main sur tout.

Fonctionnalités :	
Fonctionnalité	Description
Authentification	Inscription / Connexion sécurisée (candidat, recruteur, admin)
Création de profil	Formulaires distincts pour recruteur et candidat
Gestion des offres	CRUD des offres d'emploi par recruteur
Recherche	Recherche multi-critères (poste, lieu, type de contrat)
Postulation	Envoi de candidature avec CV/pdf
Tableau de bord	Suivi des stats (pour l'admin)
Administration	Gestion des utilisateurs et modération des annonces

Choix de la Stack Technique et Architecture

La traduction de ces besoins fonctionnels en spécifications techniques a orienté les choix vers une solution adaptée à la fois aux objectifs du projet et au contexte de formation. Le Langage back-end retenu est **PHP** avec le **framework Laravel**.

L'architecture adoptée est le Modèle-Vue-Contrôleur (**MVC**). Ce choix structurel est primordial car il garantit une séparation claire des préoccupations (logique métier, données et interface utilisateur), assurant ainsi une meilleure maintenabilité et évolutivité du code. Pour l'interface utilisateur, le **framework Vue.js** a été utilisé. Pour la persistance des données, la Base de données **MySQL** a été

sélectionnée, fonctionnant dans un environnement d'hébergement local (**XAMPP / WAMP**) pour la phase de développement.

Intégration des Exigences de Sécurité

Dès la phase de spécification, la sécurité a été intégrée comme une exigence non fonctionnelle essentielle. Les mesures prévues comprennent la protection contre les **injections SQL** et l'utilisation de **mots de passe hashés** pour sécuriser l'authentification. De plus, la gestion des **sessions sécurisées** a été définie pour protéger l'accès des utilisateurs. Ces choix initiaux sont garants de la robustesse future de la plateforme.

3. Spécifications techniques

Élément	Choix
Langage back-end	PHP
Framework Back	Laravel
Base de données	MySQL(PhpMyadmin)
Langage Front-end	HTML, CSS
Framework Front	Vue.js
Architecture	MVC (Modèle-Vue-Contrôleur)
Hébergement	Local (XAMPP / WAMP)
Sécurité	Protection contre injections SQL, mots de passe hashés, sessions sécurisées(jeton token)

1.2

Formalisation du dossier de spécifications et du planning

L'efficacité de la phase de préparation, détaillée dans la section précédente, trouve son aboutissement dans la formalisation du Dossier de Spécifications Techniques (**DST**) et l'établissement d'un Planning prévisionnel. Ces deux livrables sont essentiels pour valider l'approche choisie avec le commanditaire, ici représenté par le Directeur de campus agissant en tant que *Product Owner*.

Formalisation du Dossier de Spécifications Techniques

Le Dossier de Spécifications Techniques est le document de référence qui scelle l'accord entre les attentes fonctionnelles du client et la solution technique proposée. Pour ce projet de **Plateforme de gestion d'offres d'emploi**, la formalisation a été structurée en plusieurs étapes :

1- Synthèse du Contexte et des Objectifs : Une introduction rappelant que l'objectif principal est de proposer une interface ergonomique pour la mise en relation entre recruteurs et candidats.

2- Spécifications Fonctionnelles Détaillées : Un tableau récapitulatif a été produit, détaillant les rôles spécifiques (Administrateur, Recruteur, Candidat) et les fonctionnalités attendues. Celles-ci incluent l'authentification sécurisée, la gestion des offres (CRUD) par le recruteur, la recherche multicritères et la postulation avec CV/pdf par le candidat.

3- Spécifications Techniques : Cette section est la justification des choix de la *stack*. Le choix du **PHP (Laravel)** en *back-end* et de **MySQL** pour la base de données a été formalisé, tout comme l'adoption de l'**architecture MVC (Modèle-Vue-Contrôleur)**. Les exigences de sécurité (protection contre les injections SQL, mots de passe hashés) ont également été intégrées formellement à ce stade.

La phase de formalisation s'est conclue par une **demande de validation**. Le dossier complet, incluant le planning, a été présenté pour validation technique (cohérence des choix technologiques), validation fonctionnelle (couverture des attentes) et validation du réalisme du planning.

Établissement du Planning Prévisionnel

Le planning prévisionnel a été conçu pour structurer le développement sur une durée totale estimée à **18 semaines environ**. Ce découpage, présenté au client, est crucial pour fixer les attentes et permettre le suivi de l'avancement.

4. Planning prévisionnel

Tâche	Durée estimée
Analyse du besoin, rédaction des spécifications	1 mois
Maquettage	2 à 3 semaines
Développement des fonctionnalités de base (auth, offres)	2-6 semaines
Fonctionnalités secondaires (recherche, postulation)	2 à 3 semaines
Partie administrateur	2 semaines
Tests, corrections, amélioration UX	1 semaine
Présentation / soutenance / livraison	Janvier
	Total : 18 semaines espacées

Le planning s'articule logiquement, suivant les étapes du cycle de développement :

- **Phase d'analyse et de conception** : La première semaine a été consacrée à l'analyse du besoin et à la rédaction des spécifications. Elle est suivie par une phase de **maquettage** estimée entre une et deux semaines, essentielle pour valider l'ergonomie avant le codage.
- **Phase de développement du Cœur de Métier** : celle-ci représente la période la plus longue, dédiée au développement des fonctionnalités de base telles que l'authentification et la gestion des offres d'emploi (2 à 6 semaines environ). Les **fonctionnalités secondaires** (recherche, postulation) ont été planifiées pour la période suivante qui a durée environ 2 à 3 semaines.
- **Phase de Finalisation et de Sécurisation** : La période qui suit, sont les spécifiquement allouées à l'intégration de la **partie administration** et à la **sécurisation finale** de l'application (2 semaines).
- **Phase de Clôture** : La dernière période est réservée aux tests, aux corrections des bugs et à l'amélioration de l'expérience utilisateur (UX). Le projet se termine par la présentation, la soutenance et la livraison finale en janvier.

1.3

**Choix des technologies, des outils
de collaboration et du mode de
travail**

La phase de spécification a permis de déterminer la pile technologique (ou *stack*) la plus pertinente pour le développement de la plateforme de gestion d'offres d'emploi, tout en définissant une méthodologie de travail rigoureuse et adaptée au contexte de l'alternance.

Justification des Choix Technologiques

La cohérence technique est assurée par l'adoption de solutions éprouvées et largement utilisées :

Architecture et Langages Back-end :

- L'application repose sur une architecture **MVC** (**Modèle-Vue-Contrôleur**). Ce choix garantit la séparation des préoccupations : la logique métier est clairement isolée de l'affichage et de la gestion des données, ce qui facilite la maintenance et l'évolution future du code.
- Le langage **PHP** a été retenu pour le *back-end*. Il s'agit d'une technologie robuste et performante, particulièrement adaptée à un environnement d'hébergement local de type **XAMPP / WAMP**.
- Il est important de noter que le développement est réalisé avec le **framework Laravel**. Ce choix est motivé par la nécessité d'utiliser une solution robuste et reconnue de l'écosystème **PHP**, permettant de mettre en œuvre rapidement les principes de l'architecture **MVC** et de bénéficier des outils modernes d'un **framework** complet.

Gestion des Données et Front-end :

- La base de données choisie est **MySQL (phpMyAdmin)**, un système de gestion de base de données relationnelle (SGBDR) fiable, qui répond parfaitement aux besoins de stockage structuré des profils utilisateurs et des offres d'emploi. Cette bdd permet même au débutant de l'utiliser sans avoir besoin de coder.
- Côté *front-end*, l'interface utilisateur repose sur **HTML, CSS**. L'intégration de **Vue.js** a été choisie pour la partie dynamique. Bien que ce ne soit pas le **framework** que je maîtrisais le mieux au départ, j'ai bénéficié d'une aide ponctuelle qui m'a ensuite permis de monter en compétence en autonomie. Ce choix a assuré une mise en page responsive et une ergonomie optimale sur tous les supports.

Sécurité Intégrée : La sécurité n'est pas un ajout tardif, mais une spécification fondamentale. Les choix techniques intègrent dès le départ la **protection contre les injections SQL**, l'utilisation de **mots de passe hashés** et la mise en place de **sessions sécurisées**. Ces mesures sont la preuve d'une approche responsable de développement, en adéquation avec les bonnes pratiques du web.

Outils de Collaboration et Mode de Travail

Le mode de travail a été structuré pour maximiser la productivité et assurer la traçabilité.

- **Mode de travail et responsabilité** : Le projet est mené par l'étudiant en BTS en tant que **responsable du développement web (moi)**. Bien qu'étant seul développeur sur le cœur du projet, une structure de travail rigoureuse a été adoptée.
- **Outils d'Organisation et de Validation** : Le **Dossier de Spécifications Techniques** lui-même et le **Planning prévisionnel** constituent les outils de communication primordiaux. Ils ont permis une **validation formelle** avec le client interne (*Product Owner*) sur la cohérence technique et la faisabilité du découpage temporel.
- **Gestion de Version** : Pour garantir la traçabilité et l'intégrité du code tout au long des **périodes de développement**, l'utilisation d'un **système de gestion de version** (tel que Github et GitLab) est indispensable, même en contexte de développement individuel. Il permet un suivi précis des modifications et des retours en arrière en cas d'erreur.

A screenshot of a GitHub repository page for 'FrontVue'. The repository is public. At the top, there are buttons for 'Pin' and 'Watch'. Below that, a navigation bar shows 'main' (selected), 1 Branch, 0 Tags, a search bar with 'Go to file', and buttons for 'Add file' and 'Code'. The main area displays a commit history for a pull request by 'AhmedTwo' titled 'Memoire'. The commit message is 'fbff2ea · yesterday 2 Commits'. The commit details show 9 changes across 9 files, all of which are 'ajoutGit'. The files affected are .vscode, public, src, .env, .env.exemple, .gitattributes, .gitignore, and .prettierrc.json, each committed yesterday.

Les lignes de commandes me permettant de faire un dépôt depuis mon terminal sont :

- **git init** : Initialise un nouveau dépôt Git local dans le répertoire courant.
 - **git add .** : Permet d'ajouter tous les fichiers modifiés ou nouveaux de l'espace de travail à la **zone de préparation**.
 - **git commit -m "Message"** : Enregistre l'instantané dans l'historique local
 - **git remote add origin "URL"** : lie le dépôt local à l'URL d'un dépôt distant
 - **git push -u origin <branche>** : envoie tous les commits locaux vers le dépôt.
-
- **Articulation avec l'alternance** : Le mode de travail est intrinsèquement lié à l'alternance. Les choix technologiques sont synchronisés avec les modules de formation, permettant d'appliquer immédiatement les compétences théoriques (ex: architecture **MVC**,

sécurité) au projet concret, assurant ainsi une **synergie forte** entre la théorie et la pratique professionnelle.

1.4

Prise en compte de l'éco-conception, de l'accessibilité et de la sécurité

La prise en compte de l'éco-conception, de l'accessibilité et de la sécurité est une démonstration que le développeur Full Stack opère dans le respect des normes et des enjeux sociaux actuels.

Intégration de la Sécurité dès la Conception

La sécurité est l'exigence non fonctionnelle la plus critique pour une plateforme gérant des données personnelles (profils de candidats et de recruteurs). Dans le cadre de ce projet, les mesures suivantes ont été formalisées dans les spécifications techniques :

- **Sécurité des Données d'Authentification** : Les mots de passe des utilisateurs (candidats, recruteurs et administrateurs) ne sont jamais stockés en clair dans la base de données (**MySQL**). L'utilisation du **hachage** a été spécifiée pour garantir que même en cas de fuite de données, les identifiants restent protégés.
- **Gestion des Sessions** : La mise en place de sessions sécurisées via le jeton sanctum (**token**) puisque nous sommes en back **Laravel** a été planifiée pour garantir que seuls les utilisateurs authentifiés puissent accéder aux fonctionnalités et aux données qui leur sont propres (ex: un recruteur ne peut modifier et supprimer que ses propres offres).

Démarche d'Accessibilité et d'Ergonomie (UX)

L'accessibilité vise à garantir que l'application soit utilisable par le plus grand nombre, y compris les personnes en situation de handicap.

- **Conception d'Interface Utilisateur** : Les spécifications prévoient un travail Front-end basé sur **HTML** et **CSS**, le choix d'intégrer une conception simple et ergonomique dès le début est privilégié.
- **Respect des Normes du Web** : La phase de maquettage a été conçue pour intégrer les principes de base du **design universel**. Cela inclut, par exemple, le respect des contrastes de couleurs, la clarté de la typographie, et l'utilisation correcte des balises sémantiques en **HTML**, qui sont les fondations du Référentiel Général d'Amélioration de l'Accessibilité (**RGAA***). La vérification de l'accessibilité est explicitement prévue lors de la phase de **Tests, corrections, amélioration UX**.

*Le **RGAA** (Référentiel Général d'Amélioration de l'Accessibilité) désigne un ensemble de règles qui permettent de vérifier qu'un site web est accessible aux personnes handicapées, conformément à la loi française.

Considérations sur l'Éco-conception

Bien que non explicitement détaillés comme objectifs dans le dossier de spécifications, les principes de l'éco-conception ont été intégrés dans les choix d'architecture et de performance :

- **Optimisation de la Performance** : L'utilisation de l'architecture **MVC** favorise un code plus léger et plus performant, réduisant ainsi la charge de calcul sur le serveur.
- **Gestion des Ressources** : La performance des requêtes **MySQL** est importante. L'optimisation des requêtes et l'utilisation d'index ont été spécifiées pour minimiser le temps de réponse et, par conséquent, l'énergie consommée par l'hébergement (même local).
- **Front-end Dynamique (Vue.js)** : L'utilisation du *framework* **Vue.js** offre une interface utilisateur réactive et optimisée, facilitant la gestion de l'état de l'application et améliorant l'expérience utilisateur, tout en permettant un chargement rapide.

L'intégration de ces trois piliers dès la phase préparatoire confirme que le développement est abordé non seulement sous l'angle fonctionnel, mais aussi sous celui de la **qualité, de la responsabilité et de la durabilité**.

Partie 2 :

RNCP 37273 BC02 :

**Développer des interfaces Frontend
pour un site ou une application web
/ web mobile**

Le Lien Direct avec l'Utilisateur

Si la première partie (**BC01**) a établi l'architecture et les fondations, la **Partie 2** représente la face visible et interactive de l'application : le **Frontend**. Ce bloc de compétences est fondamental car il démontre la capacité du développeur à concrétiser l'ergonomie et le design spécifiés, transformant des maquettes statiques en une expérience utilisateur (**UX**) fluide et intuitive. L'interface est le premier contact de l'utilisateur avec l'application, et sa qualité technique et esthétique est un facteur décisif d'adoption.

La maîtrise du Frontend va bien au-delà de la simple intégration visuelle, elle exige une expertise dans la structuration sémantique du contenu, la gestion du style pour l'accessibilité, et la programmation dynamique pour l'interaction. C'est pourquoi j'ai pris Indeed, helloworld etc. comme exemples.

Structuration, Ergonomie et Design Adaptatif

Le premier défi de ce bloc est le **découpage** de la maquette graphique en blocs sémantiques clairs, puis leur traduction en **structure HTML/CSS**. Cette étape garantit non seulement la fidélité au design, mais assure également que le contenu est organisé logiquement, ce qui est important pour le référencement naturel (SEO) et, surtout, pour l'**accessibilité** de la plateforme.

Dans le cadre de ce projet de plateforme d'offres d'emploi, un accent particulier a été mis sur le Responsive Design. En s'appuyant sur les feuilles de style CSS, le développement a suivi une approche garantissant l'adaptation à toutes les tailles d'écran.

Cette approche garantit que l'affichage des profils, des offres et des formulaires de postulation s'adapte parfaitement à toutes les tailles d'écran (ordinateurs, tablettes, mobiles), offrant ainsi une expérience utilisateur optimale et cohérente, quel que soit le support.

Dynamisation et Interaction Côté Client

Au-delà du style et de la structure statique, le développeur doit savoir dynamiser l'expérience utilisateur. Cela implique la

manipulation de la page web pour interagir avec l'utilisateur et répondre à ses actions.

La mise en œuvre de cette interactivité est indispensable pour des fonctionnalités clés comme la validation de formulaires en temps réel, les animations (s'il y a lieu), ou l'affichage conditionnel d'éléments.

De plus, ce bloc exige la maîtrise des communications asynchrones. La capacité à développer des requêtes HTTP asynchrones est vitale pour communiquer avec le serveur (API) sans recharger la page entière. C'est ce qui permet d'offrir une expérience utilisateur fluide, par exemple lors d'une recherche d'offre d'emploi ou de la mise à jour d'un profil sans interruption de navigation.

En conclusion, la validation du bloc **RNCP 37273 BC02** prouve la compétence à créer des interfaces utilisateurs non seulement esthétiques et conformes aux spécifications, mais également **accessibles, ergonomiques et techniquement robustes**.

Elle illustre l'optimisation du code **front-end** pour la performance et une structuration professionnelle des fichiers.

C'est la base d'une expérience utilisateur réussie, qui sera complétée par la logique métier de la Partie 3.

2.1

Découpage de la maquette graphique et mise en place de la structure HTML

La concrétisation de l'interface utilisateur commence par la transformation des exigences fonctionnelles en une représentation visuelle fidèle et ergonomique, puis par son intégration technique.

Conception de la Maquette avec Figma

Le projet de refonte du site ***Portal_Job*** a fait de la phase de conception une étape critique. J'ai choisi d'utiliser **Figma** comme principal outil de conception. **Figma**, en tant que logiciel de design collaboratif en ligne, a permis de créer les **rendus filaires** (*wireframes*) du futur site, en respectant les **normes modernes** en matière de design web.



Grâce à son interface intuitive et à ses fonctionnalités avancées, Figma a été essentiel pour plusieurs aspects :

- **Rapidité et Collaboration** : Permettre un flux de travail efficace et la possibilité de partager le prototype interactif avec le *Product Owner* avant l'intégration.
- **Optimisation du Design** : L'utilisation de **plugins** (pour la gestion des grilles et des alignements, ainsi que l'intégration d'icônes) a été capitale pour garantir la précision du design et simplifier la transition vers le code.

Le prototype **Figma** a servi de référence visuelle initiale pour le développement, minimisant ainsi les aller-retour coûteux durant la phase de codage.

Il est important de noter qu'il s'agissait d'une étape de conception initiale et que, le rendu final étant ouvert aux ajustements, le site web présente des différences par rapport aux maquettes.

Traduction en Structure HTML Sémantique

Une fois le design validé via **Figma**, le travail d'intégration a commencé par le **découpage conceptuel** de la maquette en blocs sémantiques. L'objectif était de transformer l'image du design en une structure **HTML** valide et compréhensible par les navigateurs, les moteurs de recherche et les lecteurs d'écran.

- **Priorité à la Sémantique** : J'ai privilégié l'utilisation des **balises sémantiques HTML5** (comme `<header>`, `<nav>`, `<main>`, `<section>` et `<footer>`) mais aussi des balises comme (`<div>`, `` et `<p>`). Cette démarche assure que les lecteurs d'écran peuvent interpréter correctement le contenu et que l'accessibilité est intégrée dès le niveau structurel.
- **Intégration des Actifs** : Conformément aux spécifications techniques, la mise en place de la structure HTML a été immédiatement couplée à l'intégration des feuilles de style CSS. La

grille (héritée de la conception **Figma**) et les composants de base (barres de navigation, boutons) ont été implémentés à l'aide de CSS.

- **Cohérence du Contenu** : Pour la plateforme d'offres d'emploi, j'ai veillé à l'utilisation correcte des **niveaux de titres** (`<h1>` pour le titre principal de la page, `<h2>` pour les sections, `<h3>` pour les titres d'offres) afin de respecter la hiérarchie de l'information, essentielle à la fois pour l'accessibilité et le référencement naturel.

Cette étape a posé les fondations solides de l'interface, garantissant que l'application est non seulement esthétiquement conforme au prototype **Figma** (plus ou moins), mais également techniquement propre et prête pour l'étape suivante : la stylisation et l'adaptation aux écrans.

Wireframing

Rendu du maquettage

ADMIN

DASHBOARD

BONJOUR, "NAME" accueil offres compagny dashboard contact profil SE DECONNECTER

Admin Dashboard

- Home
- Users
- Offers
- Company

ID	NOM	PRENOM	EMAIL	MDP	ROLE
37	Fontaine	Lucas	lucasfontaine@company.com	Lucasmdp	company
36	Lemoine	Clara	clara.lemoine@company.com	Claramdp	company
35	Garnier	Marc	marc.garnier@company.com	Marcmdp	company

Tableau de bord Admin

Utilisateurs
14 Utilisateurs actifs
 Offres
10 Offres
 Compagnies
8 Compagnies

© 2000 - Tous droits réservés

USERS

BONJOUR, "NAME" accueil offres compagny dashboard contact profil SE DECONNECTER

Admin Dashboard

- Home
- Users
- Offers
- Company

ID	NOM	PRENOM	EMAIL	MDP	ROLE	ACTION
2	Martin	Sophie	sophie.martin@example.com	sophiemdp	client	
3	Lemoine	Plume	plume.lemoine@example.com	plumemdp	client	
5	Bernard	Luc	luc.bernard@example.com	lucmdp	client	
11	Ebekkoosal	Faycal	faycal@ebekkoosal.com	faycalmdp	admin	
30	Laurent	Elise	elise.laurent@example.com	Elisemdmp	company	

COMPANY

BONJOUR, "NAME" accueil offres compagny dashboard contact profil SE DECONNECTER

Admin Dashboard

- Home
- Users
- Offers
- Company

LOGO	NOM	SECTEUR	ACTION
	TechNova Solutions	Information Technology	
	GreenFields AgriCo	Agriculture	
	EduPro Learning	Education	
	UrbanConstruct Ltd.	Construction	
	MultiCare Plus	Healthcare	

© 2000 - Tous droits réservés

OFFERS

BONJOUR, "NAME" accueil offres compagny dashboard contact profil SE DECONNECTER

Admin Dashboard

- Home
- Users
- Offers
- Company

LOGO COMPANY	POSTE	CONTRAT	ACTION
	Ingénieur DevOps	CDD	
	Data Scientitst	CDD	
	Développeur P&P Backend	FREELANCE	
	Développeur Full Stack JavaSpring - Angular	FREELANCE	
	Développeur Front-End React - Stamp.js	CDD	

© 2000 - Tous droits réservés

36

NOS OFFRES

BONJOUR, "NAME" accueil offres compagny contact SE DECONNECTER

TOUTES LES OFFRES

LOGO COMPANY	POSTE	CONTRAT	ACTIONS
	Ingenieur DevOps	CDD	
	Data Scientist	CDD	
	Développeur PHP Backend	FREELANCE	
	Développeur Full Stack Application B2B	FREELANCE	
	Développeur Front-End React - SketchUp	CDD	

© 2000 - Tous droits réservés

AJOUT OFFRE

BONJOUR, "NAME" accueil offres compagny contact SE DECONNECTER

AJOUT D'UNE NOUVELLE OFFRE

TYPE	INSCRIPTION
MISIION	MISSION
POSTE	POSTE
TECHNOLOGIE	TECHNOLOGIE
REQUISITECHNIQUE	REQUISITECHNIQUE
DATE DECREATION	DATE DECREATION
LIEU	LIEU

© 2000 - Tous droits réservés

DETAILS OFFRE

BONJOUR, "NAME" accueil offres compagny contact SE DECONNECTER

Ingénieur DevOps

Description

Mission

Ville : Poste :
Entreprise : Contrat :
Technologies : Avantages :
Nombre de candidats : Date de création :

© 2000 - Tous droits réservés

MODIF OFFRE

BONJOUR, "NAME" accueil offres compagny contact SE DECONNECTER

MODIFIER UNE OFFRE

TYPE	INSCRIPTION
MISIION	MISSION
POSTE	POSTE
TECHNOLOGIE	TECHNOLOGIE
REQUISITECHNIQUE	REQUISITECHNIQUE
DATE DECREATION	DATE DECREATION

© 2000 - Tous droits réservés

POSTULER OFFRE

BONJOUR, "NAME" accueil offres compagny contact SE DECONNECTER

Postuler Maintenant !

NOM	PRÉNOM
TELÉPHONE	EMAIL
VILLE	CODI POSTAL
MOTIVATION	

Importez votre Lettre de Motivation :

Importez votre CV :

© 2000 - Tous droits réservés

2.2

Création des feuilles de style (CSS) et intégration du Responsive Design

Cette sous-section démontre la capacité à transformer la structure HTML établie (2.1) en une interface visuellement agréable, tout en garantissant une expérience utilisateur cohérente sur tous les dispositifs.

Création et Structuration des Feuilles de Style (CSS)

La stylisation a permis de mettre en forme le contenu, d'organiser les blocs et d'assurer le respect de l'identité visuelle de *Portal_Job*. J'ai ajusté la typographie (taille, graisse, polices) pour optimiser la lisibilité et respecter les normes d'accessibilité.

Contactez-nous

<p>Nos Informations</p> <div style="background-color: #f0f0f0; padding: 10px;"><p> Téléphone +33 1 23 45 67 89</p><p> Email seghiriahmed9@gmail.com</p><p> Horaires Lun. - Ven.: 9h-18h / Samedi: 10h-14h</p></div>	<p>Un message pour Portal_Job ?</p> <p>Nom : <input type="text"/></p> <p>Email : <input type="text"/></p> <p>Sujet : <input type="text"/></p> <p>Message : <input type="text"/></p> <p>Envoyer</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Pour maintenir un code CSS propre et évolutif, les feuilles de style ont été structurées en modules distincts, séparant par exemple :

```
> <script setup> ...
  </script>

> <template> ...
  </template>

<style scoped>
/* -----
 | STRUCTURE GÉNÉRALE
 | -----
 */
.page-container {
  font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;
  padding: 70px 0;
  background-color: #f8f9fa;
  display: flex;
  justify-content: center;
}
```

- **Le CSS de base** : Réinitialisation des styles et définition des variables globales (couleurs, polices).
- **Le CSS des composants** : Styles spécifiques aux éléments réutilisables (boutons, cartes d'offres, formulaires).
- **Le CSS des mises en page** : Organisation de la grille et des structures spécifiques à chaque page.

Cette approche modulaire, sans l'aide d'un *framework* comme Bootstrap, a permis de prouver une maîtrise moyenne de la cascade CSS et des techniques d'agencement modernes (Flexbox et/ou Grid Layout).

Mise en Œuvre du Responsive Design

Le Responsive Design est essentiel pour garantir l'accessibilité et la qualité de l'expérience utilisateur (UX), quel que soit l'appareil utilisé. Mon approche a consisté à développer d'abord la mise en page principale pour les écrans de bureau, avant d'assurer l'adaptation de l'affichage aux diverses tailles d'écran grâce aux feuilles de style conditionnelles.

Démarche Desktop First

1- Conception de la Mise en Page Principale : Le site a été initialement structuré pour offrir une expérience optimale sur grand écran (PC), conformément aux attentes traditionnelles de la navigation web. Cette phase a permis d'établir la grille principale, l'agencement des éléments complexes (comme la liste des offres d'emplois) et la navigation étendue.

2- Agencement des Blocs : J'ai utilisé les propriétés Flexbox et Gridlayout en CSS pour ordonner l'agencement des blocs et gérer la disposition générale du contenu de manière cohérente pour les écrans larges.

Utilisation des Media Queries pour l'Adaptation

Une fois la structure principale établie, j'ai utilisé les **@media queries** pour cibler les dispositifs de taille inférieure (tablettes et mobile) et ajuster le style. C'est l'étape où le site est véritablement rendu adaptatif :

```

@media (max-width: 992px) {
  .contact-content-flex {
    flex-direction: column;
    gap: 20px;
  }
}

@media (max-width: 768px) {
  .contact-card-pure {
    padding: 20px;
  }
}

@media (max-width: 480px) {
  .info-item-pure {
    flex-direction: column;
    text-align: center;
  }
  .icon {
    margin: 0 0 8px 0;
  }
}

.contact-content-flex {
  display: flex;
  gap: 30px;
}

.contact-card-pure {
  background-color: #fff;
  padding: 30px;
  border-radius: 12px;
  box-shadow: 0 8px 15px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s ease-in-out;
}

.info-item-pure {
  display: flex;
  align-items: center;
  background-color: #e9ecef;
  padding: 15px;
  border-radius: 8px;
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.05);
}

.icon {
  font-size: 2rem;
  margin-right: 15px;
  min-width: 40px;
}

```

Définition des Points de Rupture (*Breakpoints*) : Des points de rupture ont été choisis pour re définir la mise en page aux seuils critiques de largeur d'écran (par exemple, pour les tablettes et les mobiles).

Réorganisation des Composants : Pour les petits écrans, les *media queries* ont permis de modifier les propriétés CSS afin de **réorganiser les blocs** et de privilégier la lecture verticale.

Cette méthodologie garantit que l'interface est à la fois riche et fonctionnelle sur les grands écrans, tout en restant parfaitement navigable et ergonomique sur les appareils mobiles.