



REPOBLIKAN’I MADAGASIKARA

**Fitiavana-Tanindrazana-Fandrosoana**

-----------------------------------------

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR ET DE LA**

**RECHERCHE SCIENTIFIQUE**

--------------------------------

**UNIVERSITE DE MAHAJANGA**

------------------------

**Institut Supérieur des Sciences et Technologies de Mahajanga**

**(ISSTM)**

---------------------

**Mention: Sciences et Technique du Numérique et Physiques Appliquées**

**Parcours: Génie Informatique**

**-----------------------------**

Mémoire de fin d’étude en vue de l’obtention du Diplôme de

Licence Professionnelle

----------------

**THEME: CREATION D’UNE APPLICATION DE GESTION DE DISPENSAIRE**

Présenté par:

**Mr RAKOTONIAINA Tefy Nambinintsoa**

Président du Jury :

Examinateur :

Encadreur Pédagogique :

Encadreur Professionnelle :

ANNEE UNIVERSITAIRE : 2020 - 2021

REMERCIEMENTS

Présentation de L’ISSTM

Depuis la création du centre régional universitaire de Mahajanga en 1977, et aussi après la création des six universités de Madagascar en 1988, l’évolution de formation a été faible. L’année universitaire 2010-2011, un bon nombre d’Institut et d’offres de formations ont été créé au sein de l’Université de Mahajanga. L’Institut Supérieur des Sciences et Technologies de Mahajanga (ISSTM) en fait partie.

Les exigences du développement économique régional, les insuffisances des cadres techniciens intermédiaires dans les entreprises, les collectivités décentralisées et dans le but de réduire les dépenses des parents qui sont obligés d’envoyer leurs enfants hors de la province de Mahajanga, sont les principales raisons de la mise en place de la licence professionnelle en Sciences et Technologies en Génies Civil et Industriel au sein de l’ISSTM.

Les étudiants sortant de l’ISSTM sont opérationnels dans les domaines du génie thermique, du génie électrique, du génie civil et du génie informatique après avoir fini leur cursus.

Les contenus de ces formations scientifiques sont complétés d’une part par des visites d’imprégnation dans les milieux professionnels, des stages en entreprises et d’autre part les interventions de professeurs dans les domaines spécialisés du programme. Le diplôme de licence professionnel délivré par cet institut s’articule autour de quatre parcours :

➢ Sciences et technologies industrielles (STI) :

 Génie industriel

 Génie Thermique

 Génie Électrique

➢ Sciences et Technologies des Génies Civils (STGC) :

 Bâtiments ET travaux publics

➢ Sciences et Technique du Numérique et Physiques Appliques (STNPA) :

 Génie Informatique

L’organigramme des personnels au sein de L'ISSTM

4

LES ABRÉVIATIONS

Table of Contents

1. Généralité

A-1- Introduction générale

A-2- Problématique

A-3- Objectifs spécifiques

1. Matériels et Méthodes

B-I- Matériels

B-I-1- Environnement de développement (IDE)

1. Les outils nécessaires dans un IDE
2. Le Visual Studio Code

B-I-2- Utilisation du Framework PHP

1. C’est quoi un « FRAMEWORK »
2. Le Framework SYMFONY

B-I-3- Utilisation d’une bibliothèque BOOTSRAP

1. Définition
2. BOOTSTRAP

B-II- Méthodes

B-II-1- Merise

B-II-2- Base de données

1. Résultats

Réalisation:

* Utilisateurs
* Authentification
* Espace utilisateur:
* Espace visiteur
* Espace docteur
* Espace secrétaire
* Espace administrateur

1. Discussion

D-1- Difficultés rencontrés

D-2- Particularité du logiciel

1. Conclusion
2. GENERALITE

Introduction générale:

La propagation des maladies ingérable et contagieuse oblige beaucoup de personne à consulter les hôpitaux ou les cabinets médicaux. Grâce à ce situation, ces établissements de santé sont débordées de patient en consultation ce qui augmente le taux de contamination en vue des regroupements. Au niveau technique, l’enregistrement de ses patients s’annonce difficile sur le fait qu’il y manque de technologie approprié.

L’usage d’application web est l’un des moyens envisageable pour combler ses manques. Une **application web** désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l’utilisateur d’une application web n’a pas besoin de l’installer sur son ordinateur. Il lui suffit de se connecter à l’application à l’aide de son navigateur favori. La tendance actuelle est d’offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs [https://www.ideematic.com/dictionnaire-digital/application-web/]. C’est de là que s’inspire le développement de ce thème intitulé « CREATION D’UNE APPLICATION DE GESTION DE DISPENSAIRE » durant le stage de la fin d’étude en Licence qu’on a effectué à la société KAROKA. Une application qui permet d’avoir un moyen rapide de consulter les patients, de mieux organiser le travail et d’enregistrer rapidement les activités.

**PROBLEMATIQUE:**

Au sein de la société KAROKA, la conquête de nouvelles activités bénéficières est primordiale. Chaque année, le chef de service de KAROKA recrute des personnes possédant des talents pouvant bénéficier la société; le recrutement des développeurs fait partie de ces critères.

A part la conception des projets interne, ils proposent aussi des idées pouvant développer l’informatisation des autres établissements. Le développement du projet de création d’une application de gestion de dispensaire fait partis de ces idées. Dans le développement de ce projet, la problématique de l’étude s’annonce comme suit: face au plusieurs enregistrement des patients au sein des établissements de santé, comment éviter les pertes de données qui pourraient amener à la mal traitement des patients?

**Objectifs spécifique:**

Pour pallier à ces problèmes, nous allons implanter un système informatique qui va assurer la gestion de toutes les branches existantes dans les cabinets médicales. Ainsi, on va introduire un système qui permet de traiter automatiquement les données. Un système qui permet de gérer les diagnostics des patients, de gérer leur maladies y compris leur médicaments.

Au finale, ce système informatique a pour rôle de gérer la plus part des fonctions existant au sein des établissements médicales y compris les traitements et recherches des fichiers des patients.

Ces solutions seront apportées par notre application et tout en assurant la sécurité et la fiabilité des documents et des informations stockées dans les cabinets.

B-

MATERIELS ET METHODES

**B-I- Matériels:**

**B-I-1- Environnement de développement:**

**Connue sous l’abréviation IDE**, c’est un logiciel qui rassemble des outils permettant de développer d’autres logiciels tels que des applications mobiles, des logiciels pour ordinateur ou consoles de jeux, des sites web… [https://www.mobizel.com/definition-cest-quoi-un-ide/].

En programmation informatique, il fournit un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte, un simulateur, un compilateur ainsi qu'un débogueur. [https://fr.wikipedia.org/wiki/Environnement\_de\_d%C3%A9veloppement].

**B-I-1-a/ Les outils nécessaires dans un IDE:**

Comme nous avons ci-dessus, un bon environnement de développement doivent avoir ces quatre fonctions: un éditeur de texte, un simulateur, un compilateur ainsi qu’un débogueur.

* Editeur de texte: Elle est destinée à la programmation, en outre, à insérée notre code au moment du développement. Son éditeur de texte peut avoir une fonctionnalité de code intelligent, c’est-à-dire qu’il change automatiquement la coloration, génère une mise en forme et s’auto complète en fonction du code en entré.
* Simulateur: c’est un logiciel à l’intérieur d’un environnement de développement qui permettant de tester l’exécution de son logiciel.
* Compilateur: c’est celui-là qui va transformer le code source rédigé par le développeur en code binaire, composé de 0 et de 1 uniquement pour que l’ordinateur le comprenne.
* Débogueur: c’est une fonctionnalité qui nous aide à corriger les bugs. Dans le contexte Visual Studio, quand on débogue notre application, cela signifie généralement que nous exécutons notre application en y ayant attaché le débogueur (c’est-à-dire en mode Débogueur). Quand on faites cela, le débogueur fournit de nombreuses façons de voir ce que fait notre code pendant qu’il s’exécute. On peut parcourir notre code et examiner les valeurs stockées dans des variables, définir des espions sur des variables pour voir quand les valeurs changent, et d’examiner le chemin d’exécution de notre code [https://docs.microsoft.com/fr-fr/visualstudio/debugger/debugger-feature-tour?view=vs-2019].

Il existe de nombreux IDE. Certains permettent de développer pour un système d’exploitation spécifique, d’autres sont polyvalents. Voici quelques exemples d’IDE: Scilab , Visual Studio, Eclipse, NetBeans, Delphi, Code::Blocks, MonoDevelop, SharpDevelop, KDevelop, Access, C++ Builder, MATLAB, LabVIEW, Dev-C++, Qt Creator, XCode, Lazarus, WinDev, 4D, Dreamweaver, Zend Studio,…

**B-I-1-b/ Le Visual Studio Code:**

Visual Studio Code (ou VS Code) est un éditeur de code **léger** et **rapide** qui permet d’ajouter des **extensions**. Ce qui signifie que vous pouvez intégrer de nombreux langages et fonctionnalités.

Par rapport à un éditeur de texte basique, voici quelques exemples de fonctionnalités :

* **la coloration syntaxique du code** : en PHP et dans tous les langages, les mots écrits obéissent à une certaine grammaire : des mots clés propres au langage (en Symfony, par exemple, class, public, …), des variables (un libellé qui contient une valeur), des valeurs (nombres, chaînes de caractères), etc. Utiliser des couleurs différentes en fonction de cette grammaire permet d'améliorer grandement la lisibilité du code ;
* aide à **l'indentation du code** : en Java, le code est formé par des blocs qui se suivent et qui s'imbriquent. Pour mieux les distinguer, on utilise des espaces en début de ligne, un peu comme les paragraphes d'un roman. VS Code permet de reformater un code pour que ces espaces soient cohérents et mieux lisibles ;
* **l'autocomplétion** : VS Code peut deviner ce que vous souhaitez coder et le proposer, un peu comme sur un moteur de recherche Internet ;
* d'autres facilités diverses pour éditer du code, par exemple l'affichage de l'arborescence des fichiers à côté du code.
* **Fonctionnalités supplémentaires de Visual Studio Code**

Jetons maintenant un œil à quelques-unes des fonctionnalités qui font de Visual Code Studio un outil puissant et utilisable gratuitement pour développer des logiciels sur votre ordinateur personnel !

Le côté gauche du logiciel contient un **explorateur de fichiers**, une **barre de recherche**, un **débogueur** et une **option d’extension** ; ce dernier point étant particulièrement intéressant et utile.

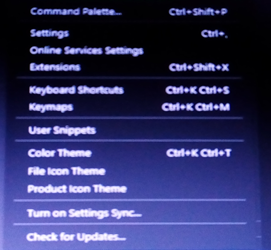
**L’onglet Extensions**

Les extensions sont de petits programmes que vous pouvez ajouter à Visual Studio Code pour **améliorer** ses fonctionnalités et le **personnaliser**. En ayant saisi du code HTML, VS Code propose de **télécharger** des extensions liées au HTML.

Commencez par cliquer sur l’icône du bas dans le panneau de gauche, appelée "Extensions". Elle mène à la marketplace des extensions qui peuvent être installées sur Visual Studio Code.

1



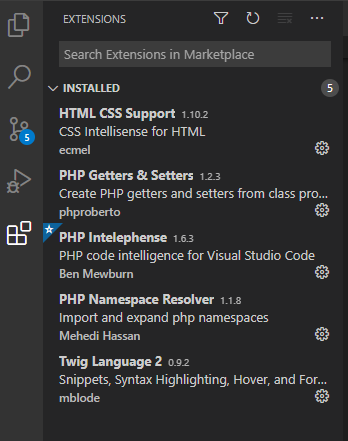


2

À gauche, vous verrez les extensions que j’ai installées pour que le langage Java soit bien supporté. Vous pouvez trouver des extensions en les tapant dans la barre de recherche en haut de la fenêtre. C’est comme pour chercher une application à installer sur votre téléphone.

Pour la recherche d’extension

Les listes des extensions qu’on a installées



**B-I-2- Utilisation du FRAMEWORK PHP:**

**B-I-2-a/ C’est quoi un FRAMEWORK?**

Un **Framework** est, comme son nom l’indique en anglais, un “**cadre de travail**“. L’objectif d’un Framework est généralement de **simplifier le travail des développeurs informatiques**, en leur offrant une **architecture “prête à l’emploi”** et qui leur permette de ne pas repartir de zéro à chaque nouveau projet [https://www.1min30.com/dictionnaire-du-web/framework].

Le Framework nous apporte beaucoup d’avantage, nous allons expliquer quelque notion en utilisant un FRAMWORK peut import si on travaille seul ou en équipe.

* **Une organisation à notre projet:**

En développant sur les FRAMEWORKS, nos projets vont être faits suivant ces étapes:

* **découpage** logique du code source
* **factorisation** de composants communs, réutilisabilité  du code
* ajout de couche **métier**
* séparation des logiques techniques / métiers (développeurs) et logiques de présentation (designeurs / intégrateurs)
* maintenance et **évolutivité**
* **Des composants et bibliothèques réutilisables [https://blog.nicolashachet.com/developpement-php/pourquoi-utiliser-un-framework-php/]:**

Bien qu’un Framework comme Symfony 2 se positionne comme **chef d’orchestre** plutôt que comme  simple agrégat de libraires, il n’empêche que les Frameworks viennent systématiquement avec leur lot de **composants réutilisables**. Que ce soit des modules, des plugins, des bundles ou tout ce qui nous fera plaisir.

* **Une incitation aux bonnes pratiques [https://blog.nicolashachet.com/developpement-php/pourquoi-utiliser-un-framework-php/]:**

C’est surtout vrai avec les **versions 2** des frameworks les plus connus (Zend, Symfony, Cake). Jusqu’à présent, chacun codait dans son coin, sans trop s’occuper des autres. Depuis quelques années, le monde PHP s’organise, se professionnalise et les développements **s’industrialisent**. Avec l’arrivée des recommandations PSR et du FIG, le code source se standardise. Les frameworks PHP actuels ne cessent de mettre en avant ces bonnes pratiques afin de rendre compatible le code issu de différents frameworks.

Ces bonnes pratiques sont le garant d’un code **lisible et compréhensible** par tous les développeurs avertis.

* **Une base régulièrement mise à jour [https://blog.nicolashachet.com/developpement-php/pourquoi-utiliser-un-framework-php/]:**

En choisissant un framework, on choisit également une **communauté** active qui détectera et corrigera des failles ou des manques du framework. En quelque sorte, on mutualise les moyens de développements. Vous profiterez ainsi des mises à jour du framework avec le lot d’améliorations (et de nouveaux bugs) qu’elles comportent.

**B-I-2-b/ Le Framework SYMFONY:**

**Symfony** est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d’accélérer le développement d'un site web [https://fr.wikipedia.org/wiki/Symfony].

La technologie Symfony intègre des mesures de sécurité innées afin de lutter contre des failles et attaques XSS, CSRF et SQL. Ce sont des failles permettant d’injecter du code HTML dans des variables mal protégées [https://www.benjaminyeurch.com/blog/developpement-web/tout-savoir-sur-symfony/].

Pour mieux développer en symfony, on a besoin d’en avoir quelque outil pour y aller rapidement:

**Terminal**: système d’exécution de tâche en ligne de commande et facilitant la génération de plusieurs autre composants;

**Git:** système pour enregistrer chaque modification de notre projet;

**Composer**: système pour gérer les dépendances;

**Symfony CLI (Command Line Interface):** système qui nous permet de pouvoir installé la commande symfony, de pouvoir utiliser HTTPS localement (site sécuriser) en exécutons la commande:

$ symfony server:ca:install

**Installation de Symfony:**

Pour installer symfony, il y a deux solutions possibles: soit avec un installer de symfony, soit avec composer. Mais avant de l’installer, il faut vérifier qu’on a la version récente de PHP (7 par exemple); sinon installons un serveur de notre choix, le lancer et de démarrer les serveurs.

* Installer symfony avec un installer symfony: on télécharge un fichier exécutable dans le site de symfony: <https://get.symfony.com/cli/setup.exe> puis l’installé (sous WINDOWS).

Sous Linux de même pour le Mac, cela se fait avec une commande:

wget https://get.symfony.com/cli/installer -O - | bash

Linux:

curl -sS https://get.symfony.com/cli/installer | bash

Mac :

* Installer symfony via composer: on télécharge un fichier exécutable dans le site: <https://getcomposer.org/Composer-Setup.exe>, on lance l’installation de  « composer » depuis le fichier télécharger. Ensuite, on exécute la commande:

composer create-project symfony/website-skeleton project\_name

Voici quelques commandes assez importantes à l’installation de symfony:

|  |  |
| --- | --- |
| Commandes | Utilisation |
| composer create-project symfony/skeleton project\_name | Pour créer un projet console ou microservice Symfony 5 en utilisant Composer. |
| symfony -V | Pour vériﬁer la version utilisée de Symfony. |
| symfony php -v | Pour connaitre la version de PHP utilisée par Symfony. |
| * cd project\_name * symfony server:start ou symfony serve | Pour lancer un projet Symfony 5 créé avec la commande symfony. |
| git clone lien\_vers\_repository\_github | cloner un projet existant |
| cd nom\_projet | se positionner dans le projet |
| composer install | installer les dépendances composer dans vendor |

Sous Visual Studio Code, l’installation de l’extension PHP Namespace Resolver nous permet d’importer les namespaces nécessaires pour certains classes.

**Structure et fonctionnement de symfony:**

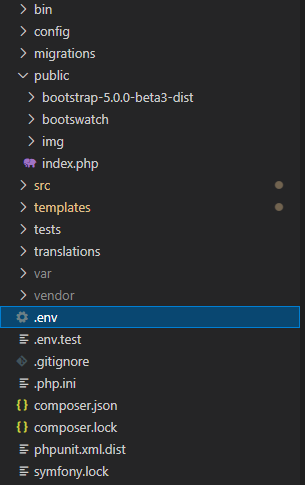
* bin/ : contenant deux exécutables, la console de Symfony et PHPUnit
* config/ : pour les ﬁchiers de conﬁguration (routes, ORM...)
* migrations/ : pour enregistrer les données de migration de la base de données
* public/ (appelé web dans la version 3) : seul dossier accessible de l’extérieur (contenant le contrôleur frontal index.php)
* src/ : pour les ﬁchiers sources de type classes (contrôleurs, entités, formulaires, DAO...)
* templates/ : contenant les vues (vue partielle) de l’application
* tests/ : pour les ﬁchiers de test
* translations/ : pour les ﬁchiers de l’internationalisation
* var/ (utilisé pendant l’exécution) : contenant les données de cache, le log, les sessions...
* vendor/ : contenant les packages nécessaires pour Symfony (mentionnés dans composer.json)

Dans le fichier « src » se trouve un fichier appelé **Kernel.php,** c’est le noyau de symfony; il est utilisé par le contrôleur frontal pour désigner le contrôleur adéquat pour répondre à la requête HTTP reçue. Le contrôleur frontal: c’est le point d’entrée d’une application symfony; il est défini dans le dossier **public/index.php.**

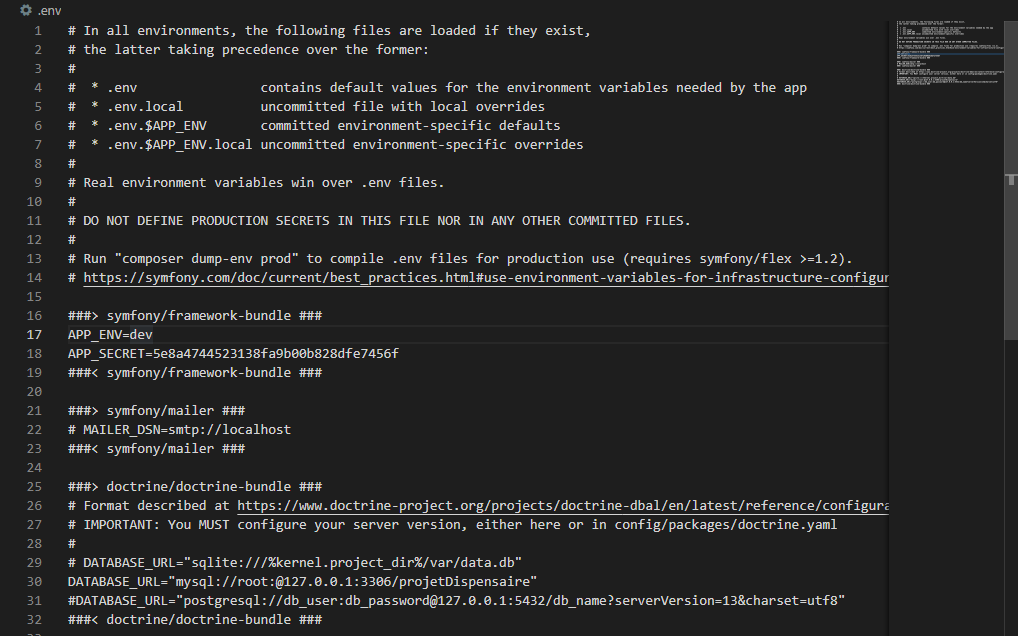
En développement symfony, il y a *deux environnements de travail*:

* **prod**: qui est destiné aux utilisateurs finaux de l’application.
* montrant l’application telle qu’elle sera visible par les visiteurs.
* rapide à exécuter
* n’affichant pas les messages d’erreur.
* **dev**: qui est destinés aux développeurs
* Plus lent que la version de production
* Environnement de débogage complet
* Possibilité d’ajouter des nouvelles fonctionnalités

Par défaut, une application en symfony est configurée à l’environnement **dev**; pour changer d’environnement, on va dans le fichier **.env** et modifier la valeur de **APP\_ENV** à **prod**.

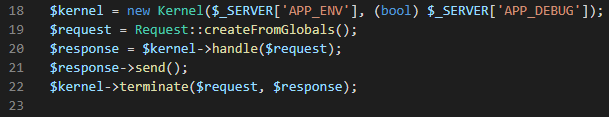


1

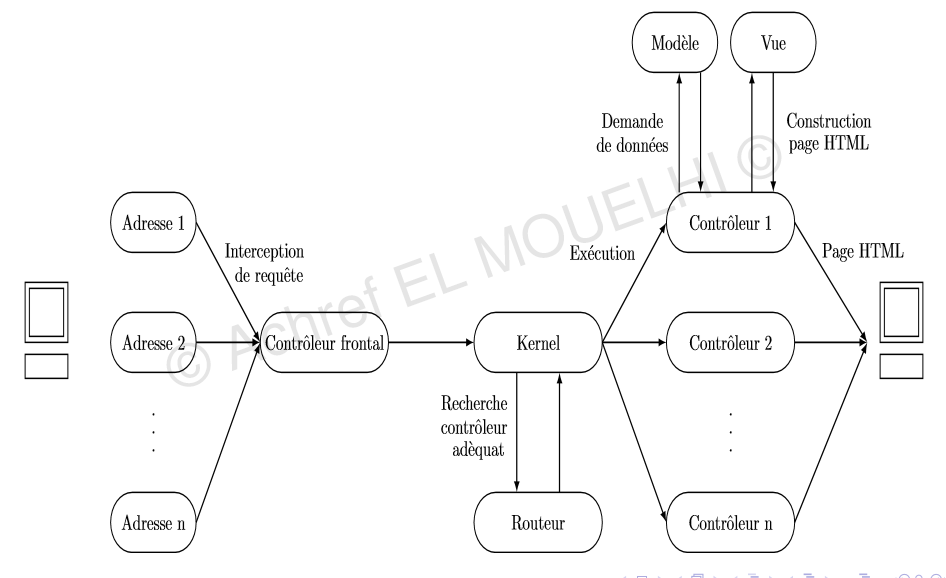


Changer en **prod**

Voici un extrait d’index.php d’un projet web de symfony:



Le contrôleur frontal charge le *kernel* selon l’environnement précisé dans **.env,** récupère la requête et utilise le *kernel* pour retourner une réponse.



1. L’utilisateur saisit l’adresse d’une page de notre site
2. Le contrôleur frontal intercepte la requête et il la transmet au Kernel
3. Le Kernel demande au Routeur le contrôleur adéquat à la page demandée
4. A la réception d’une réponse, le Kernel exécute le contrôleur
5. Le contrôleur communique avec le modèle pour récupérer ou stocker certaines données
6. Ensuite il renvoie ces données `a la vue pour qu’elle construise la page HTML et la lui retourne.
7. Enﬁn le contrôleur envoie à l’utilisateur la réponse (page HTML).