



Rapport : État d'avancement Délégué / Scrum Master

**Développement des Contrôleurs et
Interfaces des Prestataire et Client**

GR 12

29 décembre 2024

1. Introduction	3
Frontend	3
2. Interface du Prestataire	3
2.1 Tableau de Bord (Dashboard)	3
2.2 Section des Services	4
2.3 Détails d'un Service	4
2.4 Modal d'Édition de Service	5
2.5 Demandes des Clients	5
2.6 Profil Prestataire	6
2.7 Notifications :	6
Backend	7
3. Backend de Client	7
3.1. Gestion du Profil Client	7
3.2. Gestion des Services Présentés	8
3.3 Gestion des Demandes de Services	8
3.4 Gestion des Commentaires	9
3.5 Gestion des Évaluations des Services	9
3.6 Gestion des Offres	10
4. Backend de Prestataire	11
4.1 Gestion du Profil du Prestataire	11
4.2 Gestion des Certifications	11
4.3 Gestion des Demandes Clients	12
4.5 Gestion des Services Offerts	13
4.6 Gestion des Offres	14
5. Prochaines Étapes	14
5.1 Authentification et Autorisation :	14
5.2 Amélioration de l'Interface Utilisateur :	14
5.3 Extension des Fonctionnalités :	14

1. Introduction

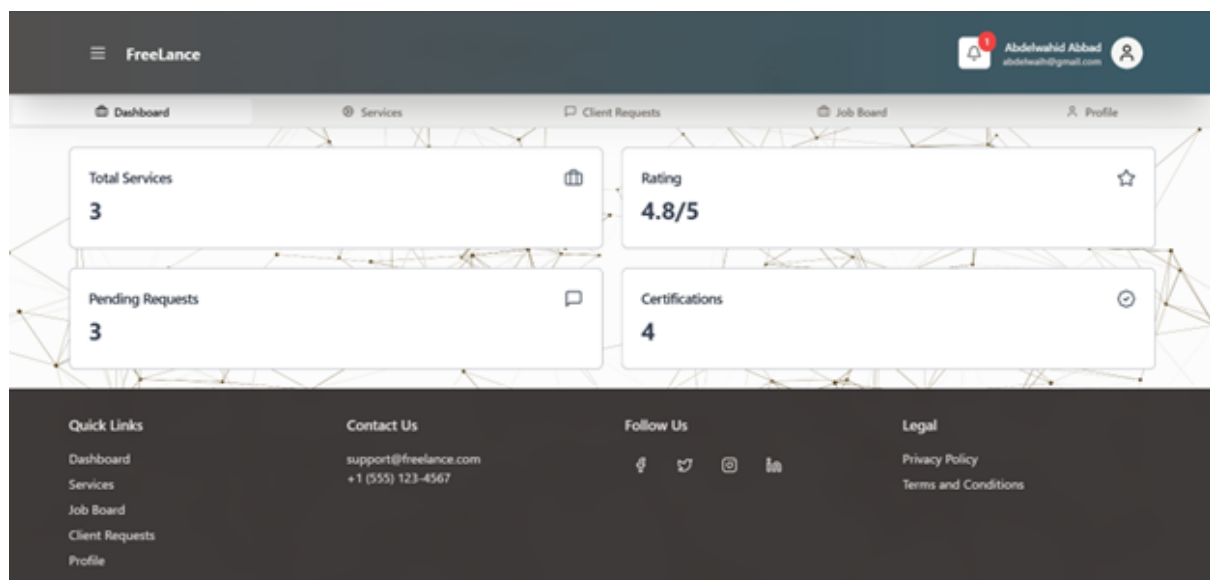
Ce rapport dresse un état d'avancement du développement des deux principaux contrôleurs de la plateforme de services : le **ClientController** et le **PrestataireController et Interfaces de prestataire**. Ces contrôleurs jouent un rôle essentiel en facilitant les interactions entre les utilisateurs (clients et prestataires) et les différentes fonctionnalités de la plateforme. Ils assurent la gestion des profils, des services, des demandes, des commentaires, des évaluations et des offres, garantissant ainsi une expérience utilisateur fluide et efficace.

Frontend

2. Interface du Prestataire

L'interface du prestataire offre une plateforme intuitive pour gérer efficacement les services, visualiser le **job board** des clients, et maintenir un profil professionnel à jour. Elle inclut également des outils pour consulter les demandes des clients et surveiller les performances globales à travers un tableau de bord clair et détaillé.

2.1 Tableau de Bord (Dashboard)



Description :

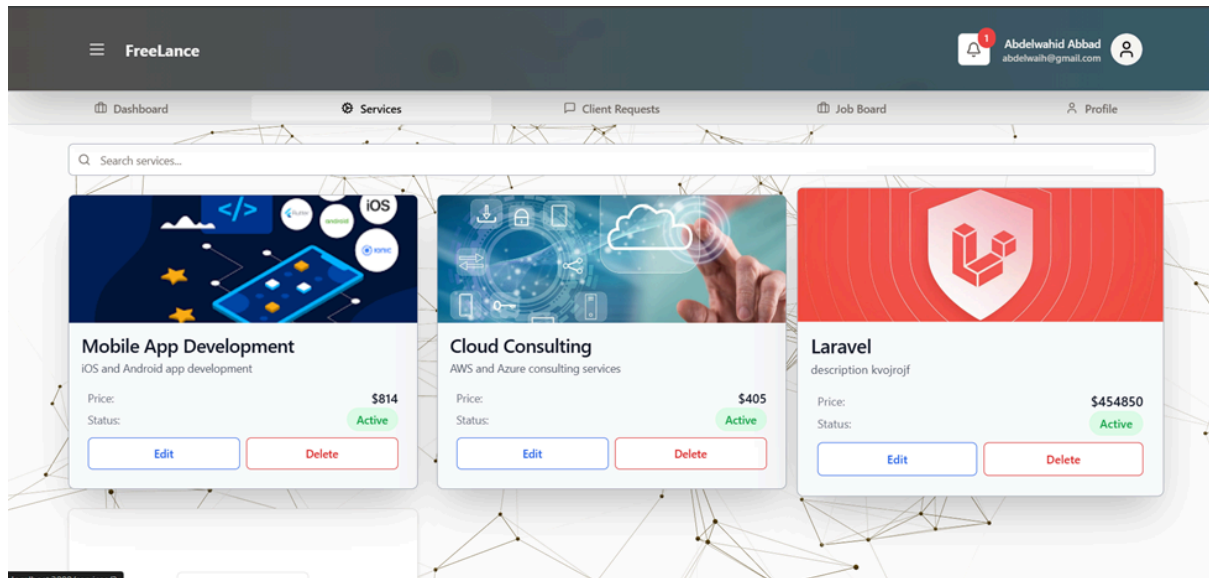
Le tableau de bord fournit un aperçu global des activités, avec des informations clés comme :

- **Services totaux** : Le nombre total de services gérés (3 dans cet exemple).
- **Demandes en attente** : Nombre de demandes en attente de traitement (3).

- **Notation** : La note moyenne des services (4.8/5).
- **Certifications** : Nombre total de certifications acquises (4).

Cette vue permet de suivre rapidement les indicateurs de performance.

2.2 Section des Services



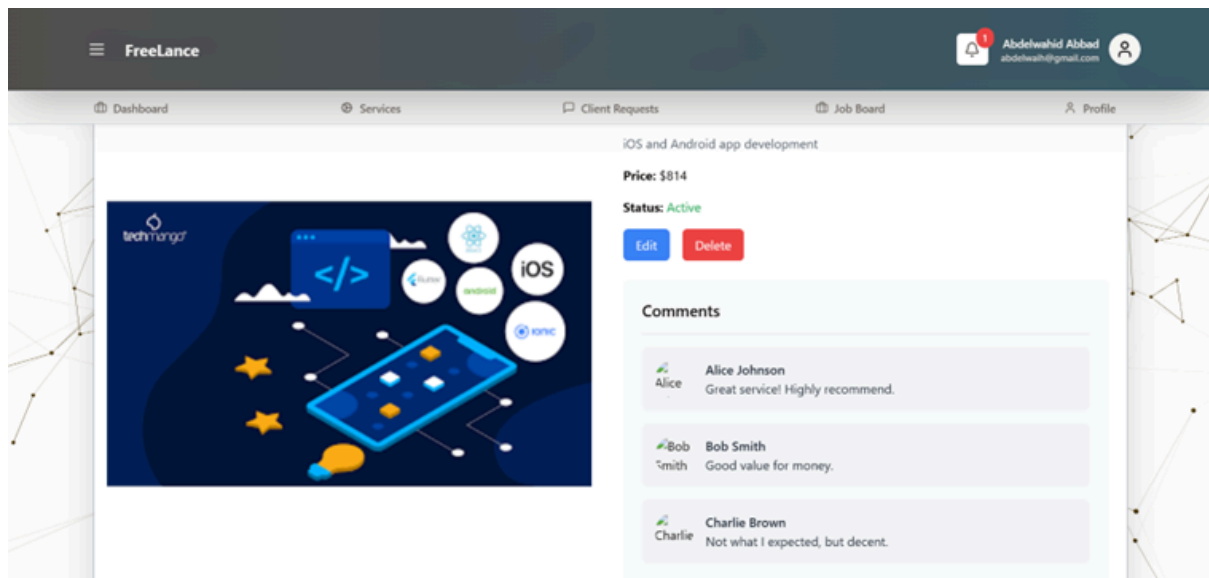
Description :

La section des services présente les services proposés par l'utilisateur, avec les détails suivants :

- **Titre et description** du service (par exemple, "Mobile App Development").
- **Prix et statut** (ex. : Actif).
- Boutons **Éditer** et **Supprimer** pour modifier ou supprimer un service.

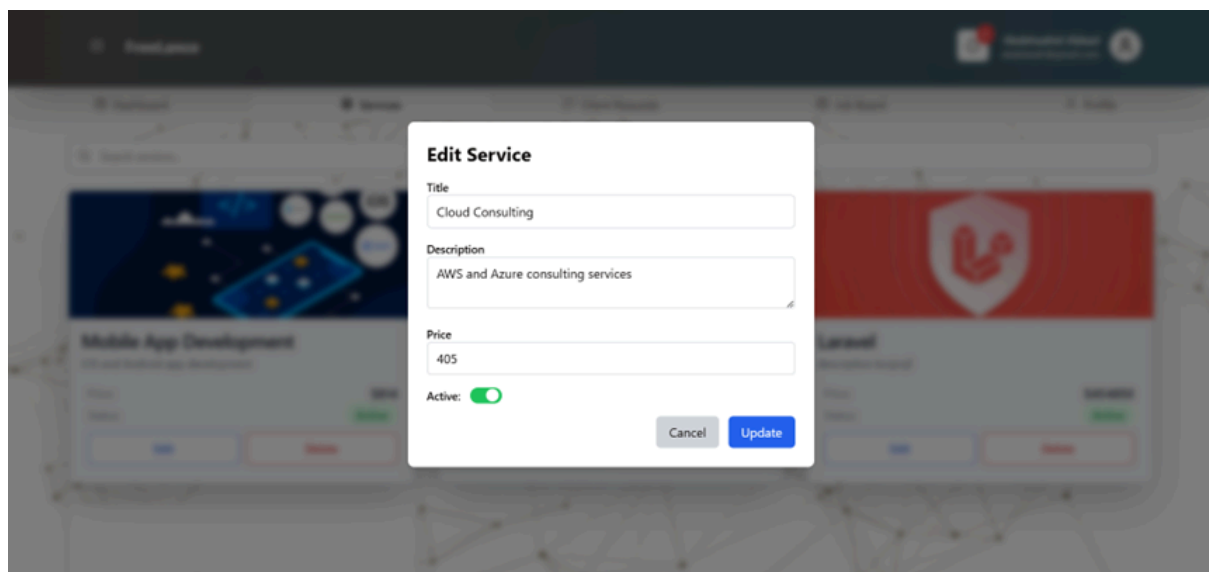
Une barre de recherche est également incluse pour faciliter la navigation.

2.3 Détails d'un Service



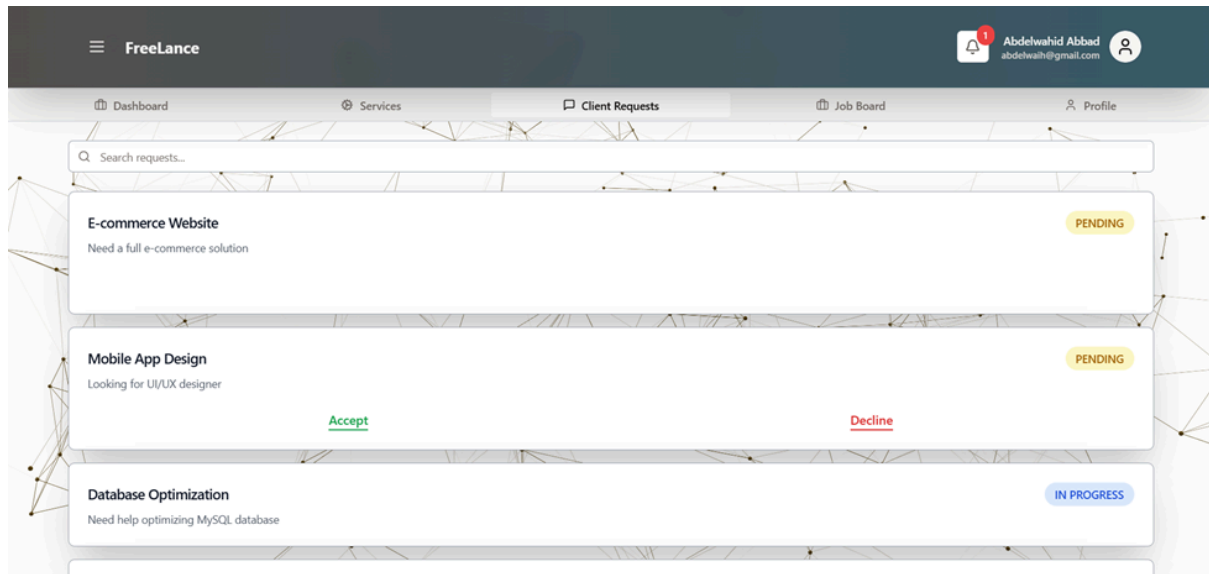
Cette page affiche les détails spécifiques d'un service

2.4 Modal d'Édition de Service



Lorsqu'un service est édité, une fenêtre modale s'affiche

2.5 Demandes des Clients



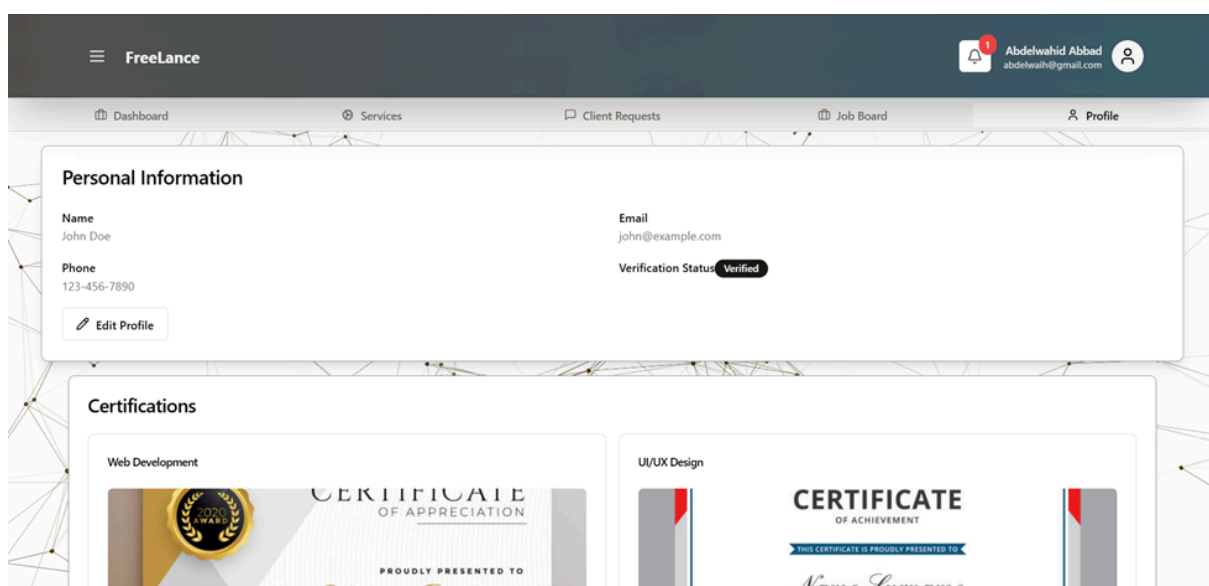
Description :

Cette section affiche les demandes des clients, organisées par statut :

- **Demande en attente** : Affichée en jaune, avec des boutons pour **Accepter** ou **Refuser**.
- **En cours** : Indiquée en bleu clair.

Chaque demande contient le titre et une brève description.

2.6 Profil Prestataire



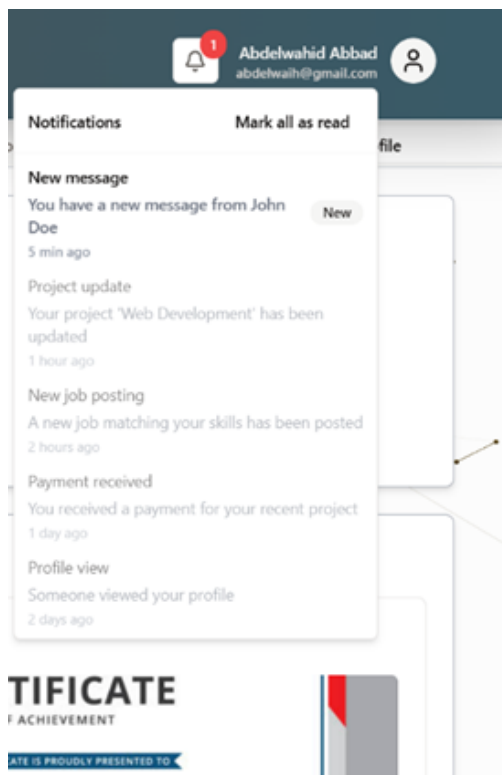
Description :

La page de profil contient :

- **Informations personnelles** : Nom, e-mail, et statut de vérification.
- **Certifications** : Une galerie des certifications acquises.

Un bouton **Modifier le profil** permet de mettre à jour les informations personnelles.

2.7 Notifications :



Le panneau de notifications offre un moyen simplifié pour le prestataire de rester informé des événements récents et des activités liées à son compte. Il inclut des options pour consulter les nouvelles notifications, les marquer comme lues, et gérer les mises à jour efficacement. Situé dans l'en-tête pour un accès rapide, il garantit que les informations importantes sont toujours à portée de main.

Backend

3. Backend de Client

Le **ClientController** est responsable de la gestion des opérations liées aux clients de la plateforme. Il permet aux clients de gérer leur profil, d'interagir avec les services proposés, de soumettre des demandes, de laisser des commentaires et des évaluations, ainsi que de gérer leurs offres.

3.1. Gestion du Profil Client

```

| | | | | /////////////// GESTION DE PROFIL/////////////////

@GetMapping("/profil")
public ClientDTO getProfil(){
    Integer idClient=2;
    return clientService.getClient(idClient);
}
@PutMapping("/profil/update")
public ResponseEntity<String> updateProfil(@RequestBody ClientDTO clientDTO){
    clientService.updateClient(clientDTO);
    return ResponseEntity.ok("Client updated");
}
@DeleteMapping("/profil/{idClient}/delete")
public ResponseEntity<String> deleteProfil(@PathVariable Integer idClient){
    clientService.deleteClient(idClient);
    return ResponseEntity.ok("Client deleted");
}

```

- **Consultation du Profil** : Permet aux clients de visualiser leurs informations personnelles.
- **Mise à Jour du Profil** : Offre la possibilité de modifier et de mettre à jour les informations personnelles.
- **Suppression du Profil** : Permet aux clients de supprimer leur compte et toutes les données associées.

3.2. Gestion des Services Présentés

```

| | | | | ///////////////GESTION DES SERVICES PRESENTEES/////////////////

//voir les categories avec les services
@GetMapping("/categories")
public List<Category> getAllCategories(){
    return categoryService.getAllCategories();
}
//voir les services pour une categorie
@GetMapping("/categories/{idCateg}/services")
public List<Service> getAllServices(@PathVariable Integer idCateg){
    return categoryService.getAllServices(idCateg);
}

```

- **Consultation des Catégories de Services** : Affiche la liste des catégories de services disponibles.

- **Consultation des Services par Catégorie** : Permet de visualiser les services spécifiques associés à une catégorie donnée.

3.3 Gestion des Demandes de Services

```

| | | | | //////////////////////////////////////////////////GERER MES DEMANDES ///////////////////////////////////
//demander une service
@PostMapping("services/{id}/demandes")
public ResponseEntity<String> demanderService(@PathVariable Integer id,@RequestBody DemandeClient demanc
    Integer idClient=2;
    demandeService.envoyerDemande(idClient,id, demandeClient);
    return ResponseEntity.ok("votre demande est envoyée");
}
//voir mes demandes

```

- **Envoi d'une Demande de Service** : Les clients peuvent soumettre des demandes pour des services spécifiques.
- **Consultation des Demandes** : Permet de visualiser et de suivre les demandes soumises.

3.4 Gestion des Commentaires

```

| | | | | /////////////// GERER MES COMMENTAIRES ///////////////
//ecrire un commentaire à propos d'une service
@PostMapping("/commentaire")
public ResponseEntity<CommentaireDTO> envoyerCommentaire(@RequestBody Commentaire commentaire) {
    Integer idClient=2;
    CommentaireDTO com= commentaireService.StoreCommentaire(idClient,commentaire);
    return ResponseEntity.status(HttpStatus.CREATED).body(com);
}
//aficher les commentaires d'une service
@GetMapping("/services/{idService}/commentaires")
public List<CommentaireDTO> getAllCommentairesofService(@PathVariable Integer idService){
    return commentaireService.getCommentairesByIdService(idService);
}

```

- **Soumission de Commentaires** : Les clients peuvent laisser des commentaires sur les services utilisés.

- **Consultation des Commentaires** : Affiche les commentaires associés à un service spécifique.

3.5 Gestion des Évaluations des Services

```

| | | | | /////////////// MES EVALUATIONS à PROPOS DES SERVICES ///////////////
//evaluer une service
@PostMapping("/services/evaluations")
public ResponseEntity<String> evaluer(@RequestBody Evaluation evaluation){
    Integer idClient=2;
    try {
        evaluationService.storeEvaluation(idClient,evaluation);
        return ResponseEntity.status(HttpStatus.CREATED).body("Évaluation enregistrée avec succès.");
    } catch (RuntimeException e) {
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(e.getMessage());
    }
}

//changer mon evaluation pour une service
@PutMapping("/services/evaluations")
public ResponseEntity<String> updateEvaluation(@RequestBody Evaluation evaluation){
    evaluationService.updateEvaluation(evaluation);
    return ResponseEntity.ok("evaluation updated");
}

//voir les evaluations d'une service
@GetMapping("/services/{idService}/evaluations")
public ResponseEntity<List<EvaluationDTO>>getAllEvaluationsOfService(@PathVariable Integer idService){
    List<EvaluationDTO> evaluations=evaluationService.getEvaluationsByServiceId(idService);
    return ResponseEntity.ok(evaluations);
}

```

- **Soumission d'Évaluations** : Permet aux clients d'évaluer les services reçus.
- **Mise à Jour des Évaluations** : Offre la possibilité de modifier les évaluations précédemment soumises.
- **Consultation des Évaluations** : Affiche les évaluations d'un service spécifique.

3.6 Gestion des Offres

```

| | | | | ///////////////MES OFFRES/////////////////
//créer un post (offre) pour chercher une service
@PostMapping("/offres")
public ResponseEntity<String> storeOffre(@RequestBody Offre offre){
    Integer idClient=2; //authentié
    offreService.createOffre(idClient,offre);
    return ResponseEntity.ok("offre stored");
}

//voir les postulas des prestataires pour une offre
@GetMapping("/offres/{id}/postulas")
public List<PostulationDTO> getAllPostulasForOffre(@PathVariable Integer id){
    return postulationService.getPostulationsForOffre(id);
}

```

- **Création d'une Offre** : Les clients peuvent créer des offres pour rechercher des services spécifiques.

- **Consultation des Postulations** : Permet de visualiser les postulations reçues des prestataires pour une offre donnée.

4. Backend de Prestataire

Le **PrestataireController** gère les opérations spécifiques aux prestataires de services. Il permet aux prestataires de gérer leur profil, leurs certifications, les demandes des clients, les services qu'ils offrent, ainsi que les offres reçues.

4.1 Gestion du Profil du Prestataire

```

////////////////////GESTION DE PROFIL //////////////////////
////////get les infos personnels de prestataire
@GetMapping("/profil")
public PrestataireProfilDTO profil() {
    Integer idPrest=1;    //le id de user authentifié
    return prestataireService.getPrestataire(idPrest);
}

//////// UPDATE les infos personnels de prestataire
@PutMapping("/profil")
public ResponseEntity<String> updateProfil(@RequestBody PrestataireProfilDTO prestataireProfilDTO) {
    Integer idPrest=1;
    prestataireService.updatePrestataire(idPrest,prestataireProfilDTO);
    return ResponseEntity.ok("Profil updated");
}

```

- **Consultation du Profil** : Permet aux prestataires de visualiser leurs informations personnelles.
- **Mise à Jour du Profil** : Offre la possibilité de modifier et de mettre à jour les informations personnelles.

4.2 Gestion des Certifications

```

| | | ////////////////////////////////////////////////// Gestion de certification////////////////////////////////////
@PostMapping("/certification/add")
public ResponseEntity<String> ajouterCertification(@RequestBody Certification certification) {
    Integer idPrest=1;
    certificationService.StoreCertification(idPrest,certification);
    return ResponseEntity.ok("Certification stored");
}

@DeleteMapping("/certification/{idCert}/delete")
public ResponseEntity<String> deleteCertification(@PathVariable Integer idCert) {
    certificationService.DeleteCertification(idCert);
    return ResponseEntity.ok("Certification deleted");
}

@PutMapping("/certification/update")
public ResponseEntity<String> updateCertification(@RequestBody Certification certification) {
    certificationService.UpdateCertification(certification);
    return ResponseEntity.ok("Certification updated");
}

```

- **Ajout d'une Certification** : Les prestataires peuvent ajouter de nouvelles certifications à leur profil.
- **Suppression d'une Certification** : Permet de supprimer des certifications existantes.
- **Mise à Jour d'une Certification** : Offre la possibilité de modifier les informations relatives à une certification.

4.3 Gestion des Demandes Clients

```

| | | | | /////////////// Gestion des demandes clients////////////////////

/////////////////voir les demandes ///////////////////
@GetMapping("/demandes")
public List<DemandeDTO> demandes() {
    Integer idPrest=1;
    return demandeService.getDemandes(idPrest);
}

///////// accepter une demande/////////
@PatchMapping("/demandes/{idDemande}/accepter")
public ResponseEntity<String> accepterDemande(@PathVariable Integer idDemande) {
    demandeService.accepterDemande(idDemande);
    return ResponseEntity.ok("le demande est acceptée!");
}

//////////annuler une demande//////////
@PatchMapping("/demandes/{idDemande}/annuler")
public ResponseEntity<String> annulerDemande(@PathVariable Integer idDemande) {
    demandeService.accepterDemande(idDemande);
    return ResponseEntity.ok("la demande est annulée");
}

```

- **Consultation des Demandes** : Permet aux prestataires de visualiser toutes les demandes de services qui leur sont adressées.
- **Acceptation d'une Demande** : Les prestataires peuvent accepter une demande spécifique.
- **Annulation d'une Demande** : Offre la possibilité d'annuler une demande en cours.

4.5 Gestion des Services Offerts

```

| | | //////////////////////////////////////////////////GESTION DES SERVICES ///////////////////////////////////
|
| //////////////////////////////////////////////////voir mes services triés selon les categories/////////////////////////////////
| @GetMapping("/mesServices")
| public List<Service> getMesServices() {
|     Integer idPrest=1;
|     return serviceService.getServices(idPrest);
| }
|
| //////////////////////////////////////////////////ajouter une service/////////////////////////////////
| @PostMapping("/mesServices/add")
| public ResponseEntity<String> addService(@RequestBody Service service) {
|     Integer idPrest=1;
|     serviceService.storeService(idPrest,service);
|     return ResponseEntity.ok("votre service est bien ajoutée");
| }
|
| //////////////////////////////////////////////////modifier une service/////////
| @PutMapping("/mesServices")
| public ResponseEntity<String> updateService(@RequestBody Service service ) {
|     serviceService.updateService(service);
|     return ResponseEntity.ok("la service est updated");
| }
|
| ////////////////////////////////////////////////// le details d'une service/////////////////////////////////
| @GetMapping("/mesServices/{idService}/serviceDetails")
| public Service afficherService(@PathVariable Integer idService) {
|     return serviceService.getService(idService);
| }
|
| //////////////////////////////////////////////////delete une service/////////////////////////////////
| @DeleteMapping("/mesServices/{idService}/delete")
| public ResponseEntity<String> deleteService(@PathVariable Integer idService) {
|     serviceService.deleteService(idService);
|     return ResponseEntity.ok("la service est deleted");
| }
|

```

- **Consultation des Services** : Permet de visualiser la liste des services proposés par le prestataire.
- **Ajout d'un Nouveau Service** : Les prestataires peuvent ajouter de nouveaux services à leur portfolio.
- **Mise à Jour d'un Service** : Offre la possibilité de modifier les détails d'un service existant.
- **Consultation des Détails d'un Service** : Permet de visualiser les caractéristiques spécifiques d'un service.
- **Suppression d'un Service** : Offre la possibilité de supprimer un service de l'offre proposée.

4.6 Gestion des Offres

