

```
1 import React, { useState } from 'react';
2 import { useParams, useNavigate } from 'react-router-dom';
3 import { useMutation, useQuery, useQueryClient } from
  '@tanstack/react-query';
4 import { Card, CardContent, CardHeader, CardTitle }
  from '@components/ui/card';
5 import { Badge } from '@components/ui/badge';
6 import { Button } from '@components/ui/button';
7 import {
8   User,
9   Mail,
10  Phone,
11  Calendar,
12  MapPin,
13  FileText,
14  ArrowLeft,
15  CreditCard,
16  Download,
17  Send,
18  RefreshCw,
19  AlertCircle
20 } from 'lucide-react';
21 import { toast } from '@hooks/use-toast';
22 import { candidatureService } from '@services/
  candidatureService';
23 import { receiptService } from '@services/
  receiptService';
24 import CandidateDocumentManager from '@components/
  admin/CandidateDocumentManager';
25 import CandidatePhotoCard from '@components/admin/
  CandidatePhotoCard';
26 import { useAdminState } from '@hooks/useAdminState'
  ;
27 import { useAdminAuth } from '@contexts/
  AdminAuthContext'; // Ajouté pour accéder au rôle
28 import { apiService } from '@services/api';
29
30 const CandidateManagement = () => {
31   const { nupcan } = useParams();
32   const navigate = useNavigate();
```

```

33  const [activeTab, setActiveTab] = useState('
    overview');
34  const queryClient = useQueryClient();
35  const { admin } = useAdminAuth(); // Récupère les
    informations de l'admin connecté
36  const { isLoading: actionLoading, executeAction
    } = useAdminState();
37
38  const { data: candidatureData, isLoading, error,
    refetch } = useQuery({
39    queryKey: ['candidature', nupcan],
40    queryFn: () => candidatureService.
    getCandidatureByNupcan(nupcan!),
41    enabled: !!nupcan,
42    retry: 3,
43    retryDelay: 1000,
44  });
45
46  const validateDocumentMutation = useMutation({
47    mutationFn: ({ documentId, statut, commentaire
    }: { documentId: number; statut: 'valide' | 'rejete'
    ; commentaire?: string }) =>
48    apiService.validateDocument(documentId.
    toString(), statut, commentaire),
49    onSuccess: () => {
50      queryClient.invalidateQueries({ queryKey: ['
    admin-dossiers'] });
51      toast({
52        title: "Document validé",
53        description: "Le candidat a été
    automatiquement notifié",
54      });
55    },
56    onError: (error) => {
57      console.error('Erreur validation:', error);
58      toast({
59        title: "Erreur",
60        description: "Impossible de valider le
    document",
61        variant: "destructive",
62      });

```

```

63     },
64   });
65
66   const handleDownloadReceipt = async () => {
67     if (!candidatureData) return;
68
69     await executeAction(
70       async () => {
71         const receiptData = {
72           candidat: {
73             ...candidatureData.candidat,
74             ldncan: candidatureData.candidat.
ldncan || 'Libreville',
75             phtcan: typeof candidatureData.
candidat.phtcan === 'string'
76               ? candidatureData.candidat.phtcan
77               : undefined
78           },
79           concours: {
80             ...candidatureData.concours,
81             fracnc: candidatureData.concours.
fracnc || 0,
82             sescnc: candidatureData.concours.
sescnc || ''
83           },
84           filiere: candidatureData.filiere ||
undefined,
85           paiement: candidatureData.paiement,
86           documents: candidatureData.documents
|| []
87         };
88         await receiptService.downloadReceiptPDF(
receiptData);
89       },
90       'Reçu téléchargé avec succès'
91     );
92   };
93
94   const handleSendReceiptEmail = async () => {
95     if (!candidatureData) return;
96

```

```

97     await executeAction(
98         async () => {
99             const receiptData = {
100                 candidat: {
101                     ...candidatureData.candidat,
102                     ldncan: candidatureData.candidat.
                        ldncan || 'Libreville',
103                     phtcan: typeof candidatureData.
                        candidat.phtcan === 'string'
104                         ? candidatureData.candidat.phtcan
105                         : undefined
106                 },
107                 concours: {
108                     ...candidatureData.concours,
109                     fracnc: candidatureData.concours.
                        fracnc || 0,
110                     sescnc: candidatureData.concours.
                        sescnc || ''
111                 },
112                 filiere: candidatureData.filiere ||
                        undefined,
113                 paiement: candidatureData.paiement,
114                 documents: candidatureData.documents
                        || []
115             };
116             await receiptService.
                generateAndSendReceiptEmail(receiptData,
                    candidatureData.candidat.maican);
117         },
118         'Reçu envoyé par email avec succès'
119     );
120 };
121
122 const handleRefresh = () => {
123     refetch();
124     toast({
125         title: "Actualisation",
126         description: "Données actualisées avec succès"
127     });
128 };

```

```

129
130   if (isLoading) {
131       return (
132           <div className="flex justify-center items-
center h-64">
133               <div className="text-center">
134                   <div className="animate-spin rounded-
full h-12 w-12 border-b-2 border-primary mx-auto mb-
4"></div>
135                   <p>Chargement des données de candidature
...</p>
136               </div>
137           </div>
138       );
139   }
140
141   if (error || !candidatureData) {
142       return (
143           <div className="text-center py-12">
144               <AlertCircle className="h-16 w-16 text-red
-500 mx-auto mb-4" />
145               <h2 className="text-xl font-semibold mb-2
text-red-600">
146                   {error ? 'Erreur de chargement' : '
Candidature non trouvée'}
147               </h2>
148               <p className="text-muted-foreground mb-4">
149                   {error ? 'Impossible de charger les
données' : 'Cette candidature n\'existe pas ou n\'
est plus accessible.'}
150               </p>
151               <div className="flex justify-center space-
x-4">
152                   <Button variant="outline" onClick=
{() => {
153                       const returnPath = admin?.role === '
admin_etablissement' ? '/admin/dashboard' : '/admin/
candidats';
154                       navigate(returnPath);
155                   }}>
156                   <ArrowLeft className="h-4 w-4 mr-2" />

```

```

157         Retour à la liste
158     </Button>
159     <Button onClick={handleRefresh}>
160         <RefreshCw className="h-4 w-4 mr-2" />
161         Réessayer
162     </Button>
163 </div>
164 </div>
165     );
166 }
167
168 const candidat = candidatureData.candidat;
169 const concours = candidatureData.concours;
170 const filiere = candidatureData.filiere;
171 const paiement = candidatureData.paiement;
172
173 return (
174     <div className="space-y-6">
175         {/* En-tête avec navigation */}
176         <div className="flex items-center justify-
177 between">
178             <div className="flex items-center space-x-
179 4">
180                 <Button variant="ghost" onClick={() => {
181                     const returnPath = admin?.role === '
182 admin_etablissement' ? '/admin/dashboard' : '/admin/
183 candidats';
184                     navigate(returnPath);
185                 }}>
186                 <ArrowLeft className="h-4 w-4 mr-2" />
187                 Retour à la liste
188             </Button>
189             <div>
190                 <h1 className="text-2xl font-bold">
191 Gestion de la Candidature</h1>
192                 <p className="text-muted-foreground">
193 NUPCAN: {nupcan}</p>
194             </div>
195         </div>
196     </div>
197     <div className="flex items-center space-x-
198 2">

```

```

191         <Button variant="outline" size="sm"
onClick={handleRefresh}>
192             <RefreshCw className="h-4 w-4 mr-2" />
193             Actualiser
194         </Button>
195     </div>
196 </div>
197
198     {/* Onglets de navigation */}
199     <div className="border-b border-gray-200">
200         <nav className="-mb-px flex space-x-8">
201             <button
202                 onClick={() => setActiveTab('
overview')}}
203                 className={`py-2 px-1 border-b-2
font-medium text-sm ${
204                     activeTab === 'overview'
205                     ? 'border-primary text-
primary'
206                     : 'border-transparent text-
gray-500 hover:text-gray-700 hover:border-gray-300'
207                 }`>
208                 >
209                 Vue d'ensemble
210             </button>
211             <button
212                 onClick={() => setActiveTab('
documents')}}
213                 className={`py-2 px-1 border-b-2
font-medium text-sm ${
214                     activeTab === 'documents'
215                     ? 'border-primary text-
primary'
216                     : 'border-transparent text-
gray-500 hover:text-gray-700 hover:border-gray-300'
217                 }`>
218                 >
219                 Documents
220             </button>
221         </nav>
222     </div>

```

```

223
224         {/ * Contenu selon l'onglet actif */}
225         {activeTab === 'overview' && (
226             <div className="grid grid-cols-1 lg:grid
- cols-3 gap-6">
227                 {/ * Photo du candidat */}
228                 <CandidatePhotoCard candidat={{
229                     nomcan: candidat.nomcan,
230                     prncan: candidat.prncan,
231                     phtcan: typeof candidat.phtcan === '
string' ? candidat.phtcan : ''
232                 }} />
233
234                 {/ * Informations du candidat */}
235                 <Card>
236                     <CardHeader>
237                         <CardTitle className="flex items-
center space-x-2">
238                             <User className="h-5 w-5" />
239                             <span>Informations Personnelles
</span>
240                         </CardTitle>
241                     </CardHeader>
242                     <CardContent className="space-y-4">
243                         <div className="grid grid-cols-2
gap-4">
244                             <div>
245                                 <label className="text-sm font
-medium text-muted-foreground">Nom</label>
246                                 <p className="font-medium">{
candidat.nomcan}</p>
247                             </div>
248                             <div>
249                                 <label className="text-sm font
-medium text-muted-foreground">Prénom</label>
250                                 <p className="font-medium">{
candidat.prncan}</p>
251                             </div>
252                         </div>
253                         <div className="flex items-center
space-x-2">

```



```

254         <Mail className="h-4 w-4 text-
muted-foreground" />
255         <span>{candidat.maican}</span>
256     </div>
257     <div className="flex items-center
space-x-2">
258         <Phone className="h-4 w-4 text-
muted-foreground" />
259         <span>{candidat.telcan}</span>
260     </div>
261     <div className="flex items-center
space-x-2">
262         <Calendar className="h-4 w-4
text-muted-foreground" />
263         <span>{new Date(candidat.dtncan
).toLocaleDateString('fr-FR')}</span>
264     </div>
265     {candidat.ldnican && (
266         <div className="flex items-
center space-x-2">
267             <MapPin className="h-4 w-4
text-muted-foreground" />
268             <span>{candidat.ldnican}</
span>
269         </div>
270     )}
271     </CardContent>
272 </Card>
273
274     {/* Informations candidature */}
275     <Card>
276         <CardHeader>
277             <CardTitle className="flex items-
center space-x-2">
278                 <FileText className="h-5 w-5" />
279                 <span>Informations Candidature</
span>
280             </CardTitle>
281             </CardHeader>
282             <CardContent className="space-y-4">
283                 <div>

```

```

284         <label className="text-sm font-
medium text-muted-foreground">Concours</label>
285         <p className="font-medium">{
concours?.libcnc || 'Non défini'}</p>
286     </div>
287     <div>
288         <label className="text-sm font-
medium text-muted-foreground">Filière</label>
289         <p className="font-medium">{
filiere?.nomfil || 'Non définie'}</p>
290     </div>
291     <div>
292         <label className="text-sm font-
medium text-muted-foreground">Statut</label>
293         <div className="mt-1">
294             <Badge
295                 className={
296                     candidat.statut === '
valide' ? 'bg-green-100 text-green-800' :
297                     candidat.statut ===
'rejete' ? 'bg-red-100 text-red-800' :
298                     'bg-orange-100
text-orange-800'
299                 }
300             >
301                 {candidat.statut || 'En
attente'}
302             </Badge>
303         </div>
304     </div>
305     <div>
306         <label className="text-sm font-
medium text-muted-foreground">Date de candidature</
label>
307         <p>{candidat.created_at ? new
Date(candidat.created_at).toLocaleDateString('fr-FR'
) : 'Non définie'}</p>
308     </div>
309     <div>
310         <label className="text-sm font-
medium text-muted-foreground">Frais de concours</

```

```

310 label>
311         <p className="font-medium">
312             {!concours?.fracnc || concours
313             .fracnc === 0 || concours.fracnc === '0'
314             ? 'GRATUIT (Programme
NGORI)'
315             : `${concours.fracnc} FCFA`
316         </p>
317     </div>
318 </CardContent>
319 </Card>
320 </div>
321 }}
322
323 { /* Informations de paiement */ }
324 { activeTab === 'overview' && paiement && (
325     <Card>
326         <CardHeader>
327             <CardTitle className="flex items-
center space-x-2">
328                 <CreditCard className="h-5 w-5" />
329                 <span>Informations de Paiement</
span>
330             </CardTitle>
331             </CardHeader>
332             <CardContent>
333                 <div className="grid grid-cols-1 md:
grid-cols-4 gap-4">
334                     <div>
335                         <label className="text-sm font-
medium text-muted-foreground">Montant</label>
336                         <p className="font-medium">{
paiement.montant} FCFA</p>
337                     </div>
338                     <div>
339                         <label className="text-sm font-
medium text-muted-foreground">Méthode</label>
340                         <p className="font-medium">{
paiement.methode}</p>
341                     </div>

```

```

341             <div>
342                 <label className="text-sm font-
medium text-muted-foreground">Référence</label>
343                 <p className="font-medium">{
paielement.reference_paiement}</p>
344             </div>
345             <div>
346                 <label className="text-sm font-
medium text-muted-foreground">Statut</label>
347                 <Badge
348                     className={
349                         paielement.statut === '
valide' ? 'bg-green-100 text-green-800' :
350                         paielement.statut === '
rejete' ? 'bg-red-100 text-red-800' :
351                         'bg-orange-100
text-orange-800'
352                     }
353                 >
354                     {paielement.statut}
355                 </Badge>
356             </div>
357         </div>
358     </CardContent>
359 </Card>
360 }}
361
362     {/* Actions administratives */}
363     {activeTab === 'overview' && (
364         <Card>
365             <CardHeader>
366                 <CardTitle>Actions Administratives</
CardTitle>
367             </CardHeader>
368             <CardContent>
369                 <div className="flex flex-wrap gap-2
">
370                     <Button
371                         onClick={handleDownloadReceipt
}
372                         variant="outline"

```

```

373         disabled={actionLoading}
374     >
375         <Download className="h-4 w-4 mr-
376         2" />
377         Télécharger le reçu
378     </Button>
379     <Button
380         onClick={
381             handleSendReceiptEmail}
382         variant="outline"
383         disabled={actionLoading}
384     >
385         <Send className="h-4 w-4 mr-2"
386     />
387     Envoyer le reçu par email
388 </Button>
389 </div>
390 </CardContent>
391 </Card>
392 )}
393
394 {/* Gestion des documents */}
395 {activeTab === 'documents' && (
396     <CandidateDocumentManager
397         candidatNupcan={nupcan!}
398         candidatInfo={{
399             nom: candidat.nomcan,
400             prenom: candidat.prnican,
401             email: candidat.maican
402         }}
403         onDocumentValidated={() => refetch
404     }
405 )}
406 />
407 )}
408 </div>
409 );
410 };
411
412 export default CandidateManagement;

```