

TRAVAIL PERSONNEL INDIVIDUEL
RÉALISÉ PAR NGUYEN KELLY

OUTIL DE COLLABORATION POUR LE COLLÈGE D'EXPERT

Module validation et répartition des TPIS

MAITRE D'APPRENTISSAGE
COMMINOT PASCAL



TPI DU 25 MAI AU 9 JUIN 2020

Table des matières

1	Arborescence des fichiers	5
2	Index	7
2.1	index.php	7
3	Modèles	8
3.1	model/checkValidation.php	8
3.2	model/connectDB.php	12
3.3	model/constDB.php	13
3.4	model/crudParams.php	14
3.5	model/crudTPIS.php	15
3.6	model/crudUser.php	17
3.7	model/crudValidation.php	18
3.8	model/crudWishes.php	20
3.9	model/displayPDF.php	22
3.10	model/displayTPIValidation.php	23
3.11	model/expertWishes.php	24
3.12	model/formParams.php	25
3.13	model/login.php	26
3.14	model/logout.php	27
3.15	model/printValidation.php	28
3.16	model/selectExpert.php	31
3.17	model/tabTPI.php	32
4	Vues	35
4.1	view/editValidation.php	35
4.2	view/footer.php	36
4.3	view/formParams.php	37
4.4	view/home.php	38
4.5	view/login.php	39
4.6	view/navH.php	40
4.7	view/navV.php	41
4.8	view/selectExpert.php	42
4.9	view/tabTPI.php	43
4.10	view/validateTPIs.php	45
5	Controlleurs	46
5.1	controller/checkValidation.php	46
5.2	controller/chooseExpert.php	47
5.3	controller/displayPDF.php	48
5.4	controller/displayTPI.php	49

5.5	controller/displayValidationTPI.php	50
5.6	controller/editParam.php	51
5.7	controller/home.php	52
5.8	controller/login.php	53
5.9	controller/logout.php	54

Listings

2.1	index.php	7
3.1	model/checkValidation.php	8
3.2	model/connectDB.php	12
3.3	model/constDB.php	13
3.4	model/crudParams.php	14
3.5	model/crudTPIS.php	15
3.6	model/crudUser.php	17
3.7	model/crudValidation.php	18
3.8	model/crudWishes.php	20
3.9	model/displayPDF.php	22
3.10	model/displayTPIValidation.php	23
3.11	model/expertWishes.php	24
3.12	model/formParams.php	25
3.13	model/login.php	26
3.14	model/logout.php	27
3.15	model/printValidation.php	28
3.16	model/selectExpert.php	31
3.17	model/tabTPI.php	32
4.1	view/editValidation.php	35
4.2	view/footer.php	36
4.3	view/formParams.php	37
4.4	view/home.php	38
4.5	view/login.php	39
4.6	view/navH.php	40
4.7	view/navV.php	41
4.8	view/selectExpert.php	42
4.9	view/tabTPI.php	43
4.10	view/validateTPIs.php	45
5.1	controller/checkValidation.php	46
5.2	controller/chooseExpert.php	47
5.3	controller/displayPDF.php	48
5.4	controller/displayTPI.php	49
5.5	controller/displayValidationTPI.php	50
5.6	controller/editParam.php	51
5.7	controller/home.php	52
5.8	controller/login.php	53
5.9	controller/logout.php	54

Chapitre 1

Arborescence des fichiers

```
/www
├── model
│   ├── checkValidation.php
│   ├── connectDB.php
│   ├── constDB.php
│   ├── crudParams.php
│   ├── crudTPIS.php
│   ├── crudUser.php
│   ├── crudValidation.php
│   ├── crudWishes.php
│   ├── displayPDF.php
│   ├── displayTPIValidation.php
│   ├── expertWishes.php
│   ├── formParams.php
│   ├── login.php
│   ├── logout.php
│   ├── printValidation.php
│   ├── selectExpert.php
│   └── tabTPI.php
├── view
│   ├── editValidation.php
│   ├── footer.php
│   ├── formParam.php
│   ├── home.php
│   ├── login.php
│   ├── navH.php
│   ├── navV.php
│   ├── selectExpert.php
│   ├── tabTPI.php
│   └── validateTPIs.php
└── controller
    ├── checkValidation.php
    ├── chooseExpert.php
    ├── displayPDF.php
    ├── displayTPI.php
    ├── displayValidationTPI.php
    └── editParam.php
```

```
graph TD; Root[ ] --- home.php; Root --- login.php; Root --- logout.php; Root --- index.php;
```

A file tree diagram showing a root directory with four files: `home.php`, `login.php`, `logout.php`, and `index.php`. The files are listed vertically, each preceded by a horizontal line segment and a vertical line segment connecting them to a common root point on the left.

Chapitre 2

Index

2.1 index.php

```
1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 session_start();
8
9 if (!empty($_SESSION['rights'])) {
10
11     $action = filter_input(INPUT_GET, "action", FILTER_SANITIZE_STRING);
12
13     if (isset($_SESSION['roles']))
14         $role = $_SESSION['roles'];
15     else
16         $role = "Nobody";
17
18     if (!in_array($action, $_SESSION['rights'][0])) {
19         $action = "home";
20     }
21
22     try {
23         require './controller/'. $action .'.php';
24     } catch (Exception $e) {
25
26     }
27 } else {
28     try {
29         require './controller/login.php';
30     } catch (Exception $e) {
31
32     }
33 }
```

Listing 2.1 – index.php

Chapitre 3

Modèles

3.1 model/checkValidation.php

```
1 <?php
2 require __DIR__ . '/../vendor/autoload.php';
3
4 use Spipu\Html2Pdf\Html2Pdf;
5 use Spipu\Html2Pdf\Exception\Html2PdfException;
6 use Spipu\Html2Pdf\Exception\ExceptionFormatter;
7
8 $tpiChoosen = FILTER_INPUT(INPUT_GET, 'tpiID', FILTER_SANITIZE_NUMBER_INT);
9 $btn = FILTER_INPUT(INPUT_POST, 'valid', FILTER_SANITIZE_STRING);
10 $assignedExpert = FILTER_INPUT(INPUT_GET, 'expert', FILTER_SANITIZE_STRING);
11 $error = '';
12
13 if (in_array($_SESSION['id'], getTPIsByld($tpiChoosen)[0]) || in_array('Administrator', $_SESSION['roles']
14 [0]) || in_array('Manager', $_SESSION['roles'][0])) {
15
16     $tpiInfo = getTPIInfoCandidate($tpiChoosen);
17
18     $expert1Name = $tpiInfo['expert1LastName'] . ' ' . $tpiInfo['expert1FirstName'];
19     $expert2Name = $tpiInfo['expert2LastName'] . ' ' . $tpiInfo['expert2FirstName'];
20
21     $validation_criteria = array(
22         "Le nombre d'heures estimés est en accord avec le règlement. (70-90 heures)",
23         "L'énoncé est entièrement rédigé par le supérieur du candidat.",
24         "L'énoncé comprend uniquement des éléments connus du candidat. (Pas de travail de recherche)",
25         "L'énoncé correspond à l'éventail normal des tâches de cette place de travail.",
26         "L'énoncé décrit clairement et sans équivoque les éléments vérifiables et de contrôle.",
27         "L'énoncé ne laisse pas de possibilité à interprétation du travail final. (Pas de phrase au
28 conditionnel)",
29         "L'énoncé décrit clairement les buts à atteindre dans le travail final.",
30         "L'aspect capital de l'énoncé fait partie du domaine professionnel selon le règlement.",
31         "L'énoncé est soluble par un collaborateur de qualification moyenne.",
32         "Le travail est exécutable individuellement. Si c'est un travail de groupe, les tâches de chaque
33 membre sont clairement définies.",
34         "Le travail de comprend pas de production de série (de manière à atteindre le minimum d'heures
35 obligatoire).",
36         "Les questions A14-A20 sont remplies.",
37         "Les questions A14-A20 sont différentes des questions de la grille d'évaluation.",
38         "Le supérieur et le candidat sont préparé à cet examen"
39     );
40
41 if (!empty(getValidation($tpiChoosen))) {
42     $tabValidation = getValidation($tpiChoosen)[0];
43     $criteria = explode(';', $tabValidation['criteria']);
44     $comment = $tabValidation['comment'];
45     $expert1Sign = $tabValidation['expert1Signature'];
46     $expert2Sign = $tabValidation['expert2Signature'];
47 } else {
48     for ($i = 0; $i < count($validation_criteria); $i++) {
49         $criteria[] = '';
50     }
51     $comment = '';
52     $expert1Sign = '';
53     $expert2Sign = '';
54 }
55
56 $form = <<<FORMVALID
57 <form action="#" method="POST">
58 <table class="table table-bordered mt-2">
59 <thead>
60 <th>Critère</th>
```



```

57 | <th>Rempli</th>
58 | </thead>
59 | FORMVALID;
60 |
61 | foreach ($validation_criteria as $key => $value) {
62 |     $form .= "<tr><td>". $value . "</td>";
63 |     $form .= "<td><select class='form-control' id='answer' name='answer' . $key . '>";
64 |     $form .= "<option></option>";
65 |
66 |     $form .= "<option ";
67 |     $form .= $criteria[$key] == 'oui' ? 'selected' : '';
68 |     $form .= ">oui</option>";
69 |
70 |     $form .= "<option ";
71 |     $form .= $criteria[$key] == 'non' ? 'selected' : '';
72 |     $form .= ">non</option>";
73 |
74 |     $form .= "<option ";
75 |     $form .= $criteria[$key] == 'n/a' ? 'selected' : '';
76 |     $form .= ">n/a</option>";
77 |
78 |     $form .= "</select></td>";
79 | }
80 | // @TODO get the name of the 2 experts
81 | $form .= <<<FORMVALID
82 | </tr>
83 | </table>
84 | <div class="form-group">
85 | <label for="comment">Commentaire</label>
86 | <textarea style="resize: none;" cols="100" class="form-control" id="comment" name="comment">$comment</
87 | textarea>
88 | </div>
89 | <tr>
90 | $expert1Name : $expert1Sign <br>
91 | $expert2Name : $expert2Sign
92 | </tr>
93 | FORMVALID;
94 |
95 | if (in_array('validateTPIs', $_SESSION['rights'][0])) {
96 |     $form .= "<a href='\"?action=editParam&tpiID=$tpiChoosen\"'><button name='valid' value='valid' class='\"
97 |     btn btn-success float-right>Valider</button></a></form>";
98 | }
99 |
100 | if ($btn == 'valid') {
101 |     $tabNewCrit = array();
102 |     $newComment = FILTER_INPUT(INPUT_POST, 'comment', FILTER_SANITIZE_STRING);
103 |
104 |     for ($i = 0; $i < count($validation_criteria); $i++) {
105 |         $crit = FILTER_INPUT(INPUT_POST, 'answer' . $i, FILTER_SANITIZE_STRING);
106 |         array_push($tabNewCrit, $crit);
107 |     }
108 |
109 |     $newCrit = implode(';', $tabNewCrit);
110 |
111 |     if (empty($criteria[0])) {
112 |         updCriteria($tpiChoosen, $newCrit);
113 |     }
114 |     else {
115 |         addCrit($tpiChoosen, $newCrit);
116 |     }
117 |
118 |     updComment($tpiChoosen, $newComment);
119 |
120 |     if (in_array('non', $tabNewCrit)) {
121 |         updStatus($tpiChoosen, 'draft');
122 |         updCriteria($tpiChoosen, $newCrit);
123 |         updExpertSign($tpiChoosen, '', 'expert1Signature');
124 |         updExpertSign($tpiChoosen, '', 'expert2Signature');
125 |     }
126 |     else {
127 |         updStatus($tpiChoosen, 'submitted');
128 |         if ($assignedExpert == '1') {
129 |             updExpertSign($tpiChoosen, date('Y-m-d H:i:s'), 'expert1Signature');
130 |         }
131 |         else {
132 |             updExpertSign($tpiChoosen, date('Y-m-d H:i:s'), 'expert2Signature');
133 |         }
134 |     }
135 |
136 |     if (empty(getValidation($tpiChoosen))) {
137 |         $tabValidation = getValidation($tpiChoosen)[0];
138 |     }
139 |
140 |     if ($tabValidation['criteria'] != $newCrit) {
141 |         updExpertSign($tpiChoosen, '', 'expert1Signature');
142 |         updExpertSign($tpiChoosen, '', 'expert2Signature');
143 |         if ($assignedExpert == '1') {

```

```

141         updExpertSign($tpiChoosen, date('Y-m-d H:i:s'), 'expert1Signature');
142     } else {
143         updExpertSign($tpiChoosen, date('Y-m-d H:i:s'), 'expert2Signature');
144     }
145 }
146
147 if ($tabValidation['expert1Signature'] != '' && $tabValidation['expert2Signature'] != '') {
148     updStatus($tpiChoosen, 'valid');
149
150     $tpiInfo = getTPIInfoCandidate($tpiChoosen);
151     $sign = getSignExpert($tpiChoosen);
152
153     $year = $tpiInfo['year'];
154     $title = $tpiInfo['title'];
155     $domain = $tpiInfo['cfcDomain'];
156     $dateStart = explode(' ', $tpiInfo['sessionStart']);
157     $dateEnd = explode(' ', $tpiInfo['sessionEnd']);
158
159     $start = $dateStart[0];
160     $end = $dateEnd[0];
161     $hourStart = explode(':', $dateStart[1])[0] . ':' . explode(':', $dateStart[1])[1];
162     $hourEnd = explode(':', $dateEnd[1])[0] . ':' . explode(':', $dateEnd[1])[1];
163
164     $workplace = $tpiInfo['workplace'];
165
166     $candLastName = $tpiInfo['candidateLastName'];
167     $candFirstName = $tpiInfo['candidateFirstName'];
168     $candPhone = $tpiInfo['candidatePhone'];
169     $candMail = $tpiInfo['candidateMail'];
170
171     $managerCompagny = $tpiInfo['managerCompagny'];
172     $managerLastName = $tpiInfo['managerLastName'];
173     $managerFirstName = $tpiInfo['managerFirstName'];
174     $managerPhone = $tpiInfo['managerPhone'];
175     $managerMail = $tpiInfo['managerMail'];
176
177     $expert1LastName = $tpiInfo['expert1LastName'];
178     $expert1FirstName = $tpiInfo['expert1FirstName'];
179     $expert1Phone = $tpiInfo['expert1Phone'];
180     $expert1Mail = $tpiInfo['expert1Mail'];
181
182     $expert2LastName = $tpiInfo['expert2LastName'];
183     $expert2FirstName = $tpiInfo['expert2FirstName'];
184     $expert2Phone = $tpiInfo['expert2Phone'];
185     $expert2Mail = $tpiInfo['expert2Mail'];
186
187     $expert1Sign = explode(' ', $sign['expert1Signature']);
188     $expert1SignDate = $expert1Sign[0];
189     $expert1HourSign = explode(':', $expert1Sign[1])[0] . ':' . explode(':', $expert1Sign[1])[1];
190     $expert1Valid = "$expert1LastName $expert1FirstName : Validé le $expert1SignDate à
191 $expert1HourSign";
192
193     $expert2Sign = explode(' ', $sign['expert2Signature']);
194     $expert2SignDate = $expert2Sign[0];
195     $expert2HourSign = explode(':', $expert2Sign[1])[0] . ':' . explode(':', $expert2Sign[1])[1];
196     $expert2Valid = "$expert2LastName $expert2FirstName : Validé le $expert2SignDate à
197 $expert2HourSign";
198
199     $comment = $tabValidation['comment'];
200
201     $format = 'Validation_TPI_' . $year . '_' . $tpiChoosen . '_' . $candLastName . '_' .
202     $candFirstName . '.pdf';
203     updPdfPath($tpiChoosen, $format);
204
205     try {
206         ob_start();
207         require_once 'printValidation.php';
208         $exp = ob_get_clean();
209
210         $html2pdf = new Html2Pdf('P', 'A4', 'fr');
211         $html2pdf->pdf->SetDisplayMode('fullpage');
212         $html2pdf->writeHTML($exp);
213         $html2pdf->output(__DIR__ . '/../pdf/' . $format, 'F');
214     } catch (Html2PdfException $e) {
215         $html2pdf->clean();
216
217         $formatter = new ExceptionFormatter($e);
218         echo $formatter->getHtmlMessage();
219     }
220 }
221
222 header('Location: ?action=displayValidationTPI');
223

```

```
224|         exit;
225|     }
226| } else {
227|     // header('Location: ?action=home');
228|     // exit;
229| }
```

Listing 3.1 — model/checkValidation.php

3.2 model/connectDB.php

```
1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 require_once 'constDB.php';
8 /**
9  * Do the connection to the database
10 * @return PDO connection of the database
11 */
12 function getConnexion()
13 {
14     static $db = null;
15
16     if ($db === null) {
17         try {
18             $connexionString = 'mysql:host=' . DB_HOST . ';dbname=' . DB_NAME . '';
19             $db = new PDO($connexionString, DB_USER, DB_PASS);
20             $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
21         } catch (PDOException $e) {
22             die('Erreur : ' . $e->getMessage());
23         }
24     }
25     return $db;
26 }
```

Listing 3.2 – model/connectDB.php

3.3 model/constDB.php

```
1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 DEFINE( 'DB_HOST', 'localhost' );
8 DEFINE( 'DB_NAME', 'tpidb' );
9 DEFINE( 'DB_USER', 'tpi' );
10 DEFINE( 'DB_PASS', 'Super' );
```

Listing 3.3 – model/constDB.php

3.4 model/crudParams.php

```
1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 /**
8  * update the value by the name given
9  * @param string name of the param
10 * @param string value of the param
11 */
12 function updParam($name, $value) {
13     $upd = getConnexion();
14     $req = $upd->prepare("UPDATE params SET value = :value WHERE name = :name");
15     $req->bindParam(":name", $name, PDO::PARAM_STR);
16     $req->bindParam(":value", $value, PDO::PARAM_STR);
17     $req->execute();
18 }
19
20 /**
21 * get all the params
22 */
23 function getParams() {
24     $upd = getConnexion();
25     $req = $upd->prepare("SELECT name, value FROM params");
26     $req->execute();
27     return $res = $req->fetchAll(PDO::FETCH_KEY_PAIR);
28 }
29
30 /**
31 * get the param by the name
32 * @param string name of the value
33 */
34 function getParamsByName($name) {
35     $upd = getConnexion();
36     $req = $upd->prepare("SELECT value FROM params WHERE name = :name");
37     $req->bindParam(':name', $name, PDO::PARAM_STR);
38     $req->execute();
39     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
40 }
```

Listing 3.4 – model/crudParams.php

3.5 model/crudTPIS.php

```

1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 require_once 'connectDB.php';
8
9 /**
10 * update the userExpert1ID by the ID given
11 * @param int tpiID
12 * @param int expertID
13 */
14 function updTPIExp1($tpiID, $expertID) {
15     $upd = getConnexion();
16     $req = $upd->prepare("UPDATE 'tpis' SET 'userExpert1ID' = :expert1 WHERE 'tpis'.'. 'tpiID' = :tpiID");
17     $req->bindParam(":tpiID", $tpiID, PDO::PARAM_INT);
18     $req->bindParam(":expert1", $expertID, PDO::PARAM_INT);
19     $req->execute();
20 }
21
22 /**
23 * update the userExpert1ID by the ID given
24 * @param int tpiID
25 * @param int expertID
26 */
27 function updTPIExp2($tpiID, $expertID) {
28     $upd = getConnexion();
29     $req = $upd->prepare("UPDATE 'tpis' SET 'userExpert2ID' = :expert2 WHERE 'tpis'.'. 'tpiID' = :tpiID");
30     $req->bindParam(":tpiID", $tpiID, PDO::PARAM_INT);
31     $req->bindParam(":expert2", $expertID, PDO::PARAM_INT);
32     $req->execute();
33 }
34
35 /**
36 * update the status of the TPI
37 * @param int tpiID
38 * @param int stat of the TPI
39 */
40 function updStatus($tpiID, $stat) {
41     $upd = getConnexion();
42     $req = $upd->prepare("UPDATE 'tpis' SET 'tpiStatus' = :stat WHERE 'tpiID' = :tpiID");
43     $req->bindParam(":tpiID", $tpiID, PDO::PARAM_INT);
44     $req->bindParam(":stat", $stat, PDO::PARAM_STR);
45     $req->execute();
46 }
47
48 /**
49 * get all TPIS
50 */
51 function getTPIs(){
52     $tpi = getConnexion();
53     $sql = "SELECT tpiID, pdfPath, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd,
54 presentationDate, workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, uc.LastName AS
55 candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.
56 FirstName AS managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.
57 LastName AS expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.
58 companyName FROM tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON
59 userManagerID = um.userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2
60 ON userExpert2ID = ue2.userID WHERE year = YEAR(CURDATE())";
61     $req = $tpi->prepare($sql);
62     $req->execute();
63     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
64 }
65
66 /**
67 * get all TPIS by id
68 * @param int id of the TPI
69 */
70 function getTPIsById($id){
71     $tpi = getConnexion();
72     $sql = "SELECT tpiID, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd, presentationDate,
73 workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, pdfPath, uc.LastName AS
74 candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.
75 FirstName AS managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
76 expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName FROM
77 tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.
78 userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID =
79 ue2.userID WHERE tpiID = :id";
80     $req = $tpi->prepare($sql);
81     $req->bindParam(':id', $id, PDO::PARAM_INT);
82     $req->execute();
83     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
84 }

```

```

70 }
71
72 /**
73  * get all TPis that has experts assigned
74  */
75 function getTPisWEExpert(){
76     $tpi = getConnexion();
77     $sql = "SELECT tpiID, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd, presentationDate,
workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, pdfPath, uc.LastName AS
candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.FirstName AS
managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName FROM
tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.
userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID =
ue2.userID WHERE userExpert1ID IS NOT NULL AND userExpert2ID IS NOT NULL ORDER BY tpiID";
78     $req = $tpi->prepare($sql);
79     $req->execute();
80     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
81 }
82
83 /**
84  * get all TPis of the manager
85  * @param int id of the manager
86  */
87 function getTPisOfManager($idUser){
88     $tpi = getConnexion();
89     $sql = "SELECT tpiID, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd, presentationDate,
workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, pdfPath, uc.LastName AS
candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.FirstName AS
managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName FROM
tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.
userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID =
ue2.userID WHERE userManagerID = :idUser AND userExpert1ID IS NOT NULL AND userExpert2ID IS NOT NULL
ORDER BY tpiID";
90     $req = $tpi->prepare($sql);
91     $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
92     $req->execute();
93     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
94 }
95
96 /**
97  * get all TPis of the expert
98  * @param int id of expert
99  */
100 function getTPisOfExpert($idUser){
101     $tpi = getConnexion();
102     $sql = "SELECT tpiID, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd, presentationDate,
workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, pdfPath, uc.LastName AS
candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.FirstName AS
managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName FROM
tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.
userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID =
ue2.userID WHERE userExpert1ID = :idUser OR userExpert2ID = :idUser";
103     $req = $tpi->prepare($sql);
104     $req->bindParam(':idUser', $idUser, PDO::PARAM_INT);
105     $req->execute();
106     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
107 }
108
109 /**
110  * get TPI information of the candidate by the tpiID
111  * @param int id of the TPI
112  */
113 function getTPIInfoCandidate($id){
114     $tpi = getConnexion();
115     $sql = "SELECT tpiID, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd, presentationDate,
workplace, userCandidateID, userManagerID, userExpert1ID, userExpert2ID, uc.LastName AS
candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.FirstName AS
managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName, um.
companyName as managerCompagny, um.phone as managerPhone, um.email as managerMail, uc.phone as
candidatePhone, uc.email as candidateMail, ue1.phone as expert1Phone, ue1.email as expert1Mail, ue2.
phone as expert2Phone, ue2.email as expert2Mail, description FROM tpis LEFT JOIN users AS uc ON
userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.userID LEFT JOIN users AS ue1
ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID = ue2.userID WHERE tpiID = :id";
116     $req = $tpi->prepare($sql);
117     $req->bindParam(':id', $id, PDO::PARAM_INT);
118     $req->execute();
119     return $res = $req->fetch();
120 }

```

Listing 3.5 – model/crudTPIS.php

3.6 model/crudUser.php

```

1 <?php
2 /**
3  * Log the user with the email and check the password
4  * @param string email of the user
5  * @param string pwd of the user
6  */
7 function login($email, $pwd){
8     $log = getConnexion();
9     $req = $log->prepare("SELECT userID, lastName, firstName, email, pwdHash, pwdSalt FROM users WHERE
10     email = :email");
11     $req->bindParam(':email', $email, PDO::PARAM_STR);
12     $req->execute();
13     $res = $req->fetch();
14     //return false if the email is invalid
15     if (empty($res)) {
16         return false;
17     }
18     if ($res['pwdHash'] == SHA1($pwd . $res['pwdSalt'])) {
19         $_SESSION['roles'] = getRole($res['userID']);
20         $_SESSION['rights'] = array();
21         $rights = array();
22         foreach ($SESSION['roles'] as $key => $value) {
23             foreach (getUserRights($value['roleName']) as $right) {
24                 if (!in_array($right['rightName'], $rights)) {
25                     array_push($rights, $right['rightName']);
26                 }
27             }
28             array_push($_SESSION['rights'], $rights);
29         }
30         $_SESSION['id'] = $res['userID'];
31         $_SESSION['name'] = $res['lastName'] . ' ' . $res['firstName'];
32         return true;
33     }
34     return false;
35 }
36
37 /**
38  * get the role of the user by it ID
39  * @param int id of the user
40  */
41 function getRole($id){
42     $role = getConnexion();
43     $req = $role->prepare("SELECT roleName FROM user_roles LEFT JOIN roles AS r ON user_roles.roleID = r.
44     roleID WHERE userID = :id");
45     $req->bindParam(':id', $id, PDO::PARAM_STR);
46     $req->execute();
47     return $req->fetchAll(PDO::FETCH_ASSOC);
48 }
49
50 /**
51  * get all user rights by the roleName
52  * @param string rolename
53  */
54 function getUserRights($roleName){
55     $role = getConnexion();
56     $req = $role->prepare("SELECT rightName FROM 'role_rights' AS rr LEFT JOIN rights AS r ON rr.rightID =
57     r.rightID LEFT JOIN roles AS rl ON rr.roleID = rl.roleID WHERE roleName = :roleName");
58     $req->bindParam(':roleName', $roleName, PDO::PARAM_STR);
59     $req->execute();
60     return $req->fetchAll(PDO::FETCH_ASSOC);
61 }
62
63 }

```

Listing 3.6 – model/crudUser.php

3.7 model/crudValidation.php

```

1 <?php
2 require_once 'connectDB.php';
3
4 /**
5  * add criterions in the table tpi_validations with the tpiID
6  * @param int tpiID
7  * @param string crit is the string of criterions with a delimiter ';'
8  */
9 function addCrit($tpiID, $crit){
10     $add = getConnexion();
11     $req = $add->prepare("INSERT INTO tpi_validations (tpiID, criterions) VALUES (:tpiID, :crit)");
12     $req->bindParam(':tpiID', $tpiID, PDO::PARAM_INT);
13     $req->bindParam(':crit', $crit, PDO::PARAM_STR);
14     $req->execute();
15 }
16
17 /**
18  * get TPI validation by it ID
19  * @param int id of the TPI
20  */
21 function getValidation($id){
22     $tpi = getConnexion();
23     $req = $tpi->prepare("SELECT * FROM 'tpi_validations' WHERE tpiID = :id");
24     $req->bindParam(':id', $id, PDO::PARAM_INT);
25     $req->execute();
26     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
27 }
28
29
30 /**
31  * get TPI signature by it ID
32  * @param int id of the TPI
33  */
34 function getSignExpert($id){
35     $tpi = getConnexion();
36     $req = $tpi->prepare("SELECT expert1Signature, expert2Signature FROM tpi_validations WHERE tpiID = :id");
37     $req->bindParam(':id', $id, PDO::PARAM_INT);
38     $req->execute();
39     return $res = $req->fetch();
40 }
41
42 /**
43  * update the criterions of the TPI
44  * @param int tpiID
45  * @param string crit is the criterions of the TPI, the delimitation is ';'
46  */
47 function updCriterions($tpiID, $crit) {
48     $upd = getConnexion();
49     $req = $upd->prepare("UPDATE 'tpi_validations' SET 'criterions' = :crit WHERE 'tpiID' = :tpiID");
50     $req->bindParam(':crit', $crit, PDO::PARAM_STR);
51     $req->bindParam(':tpiID', $tpiID, PDO::PARAM_STR);
52     $req->execute();
53 }
54
55 function updComment($tpiID, $comment) {
56     $upd = getConnexion();
57     $req = $upd->prepare("UPDATE 'tpi_validations' SET 'comment' = :comment WHERE 'tpiID' = :tpiID");
58     $req->bindParam(':comment', $comment, PDO::PARAM_STR);
59     $req->bindParam(':tpiID', $tpiID, PDO::PARAM_INT);
60     $req->execute();
61 }
62
63 function updExpertSign($tpiID, $sign, $expert) {
64     $upd = getConnexion();
65     $req = $upd->prepare("UPDATE 'tpi_validations' SET $expert = :expertSign WHERE 'tpiID' = :tpiID");
66     if ($sign == '') {
67         $sign = NULL;
68         $req->bindParam(':expertSign', $sign, PDO::PARAM_NULL);
69     } else {
70         $req->bindParam(':expertSign', $sign, PDO::PARAM_STR);
71     }
72     $req->bindParam(':tpiID', $tpiID, PDO::PARAM_INT);
73     $req->execute();
74 }
75
76 function updPdfPath($tpiID, $path) {
77     $upd = getConnexion();
78     $req = $upd->prepare("UPDATE 'tpi_validations' SET 'pdfPath' = :pdfPath WHERE 'tpiID' = :tpiID");
79     $req->bindParam(':pdfPath', $path, PDO::PARAM_STR);
80     $req->bindParam(':tpiID', $tpiID, PDO::PARAM_INT);
81     $req->execute();
82 }

```

Listing 3.7 – model/crudValidation.php

3.8 model/crudWishes.php

```

1 <?php
2 /**
3  * @author Nguyen Kelly
4  * @version 1.0
5  */
6
7 /**
8  * Insert a wish in the table wishes
9  * @param int idExpert is the Id of the Expert assign to the TPI
10 * @param int tpild is the ID of the TPI
11 */
12 function addWish($idExpert, $tpild)
13 {
14     $connexion = getConnexion();
15     $req = $connexion->prepare("INSERT INTO wishes (userExpertID, tpild) VALUES (:expert, :tpild)");
16     $req->bindParam(":expert", $idExpert, PDO::PARAM_INT);
17     $req->bindParam(":tpild", $tpild, PDO::PARAM_INT);
18     $req->execute();
19 }
20
21 /**
22 * Delete a wish by its id
23 * @param int idExpert
24 * @param int id of the TPI
25 */
26 function rmWish($idExpert, $id){
27     $rm = getConnexion();
28     $req = $rm->prepare("DELETE FROM wishes WHERE userExpertID = :idExpert AND tpild = :id");
29     $req->bindParam(":idExpert", $idExpert, PDO::PARAM_INT);
30     $req->bindParam(":id", $id, PDO::PARAM_INT);
31     $req->execute();
32 }
33
34 /**
35 * get all wishes of the TPI by its ID that hasn't been assigned
36 * @param int id of the TPI
37 */
38 function getWishesByTpildAssignedNull($id){
39     $wishes = getConnexion();
40     $req = $wishes->prepare("SELECT userExpertID, lastName as expertLastName, firstName as expertFirstName
41     , 'tpild', assigned
42     FROM 'wishes' AS w
43     LEFT JOIN users AS ue1 ON w.userExpertID = ue1.userID
44     WHERE assigned IS NULL
45     AND tpild = :id");
46     $req->bindParam(':id', $id, PDO::PARAM_INT);
47     $req->execute();
48     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
49 }
50
51 /**
52 * get all wishes of the TPI by its ID that has been assigned
53 * @param int id of the TPI
54 */
55 function getWishesByTpild($id){
56     $wishes = getConnexion();
57     $req = $wishes->prepare("SELECT userExpertID, lastName as expertLastName, firstName as expertFirstName
58     , 'tpild', assigned
59     FROM 'wishes' AS w
60     LEFT JOIN users AS ue1 ON w.userExpertID = ue1.userID
61     WHERE tpild = :id
62     AND assigned IS NOT NULL");
63     $req->bindParam(':id', $id, PDO::PARAM_INT);
64     $req->execute();
65     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
66 }
67
68 /**
69 * get the wish of the user by ids
70 * @param int id of the TPI
71 */
72 function getWishUser($idExpert, $tpild){
73     $wishes = getConnexion();
74     $req = $wishes->prepare("SELECT userExpertID, tpild FROM 'wishes' WHERE userExpertID = :id AND tpild = :tpild");
75     $req->bindParam(':id', $idExpert, PDO::PARAM_INT);
76     $req->bindParam(':tpild', $tpild, PDO::PARAM_INT);
77     $req->execute();
78     return $res = $req->fetchAll(PDO::FETCH_ASSOC);
79 }
80 /**

```

```
81 | * update the value by the name given
82 | * @param string name of the param
83 | * @param string value of the param
84 | */
85 | function updWish($idExpert, $tpi, $assigned = null) {
86 |     $upd = getConnexion();
87 |     $req = $upd->prepare("UPDATE 'wishes' SET 'assigned' = :assigned WHERE 'userExpertID' = :expert AND '
      | tpiID' = :tpi");
88 |     $req->bindParam(":expert", $idExpert, PDO::PARAM_INT);
89 |     $req->bindParam(":assigned", $assigned, PDO::PARAM_INT);
90 |     $req->bindParam(":tpi", $tpi, PDO::PARAM_INT);
91 |     $req->execute();
92 | }
```

Listing 3.8 – model/crudWishes.php

3.9 model/displayPDF.php

```
1 <?php
2 $path = FILTER_INPUT(INPUT_GET, 'link', FILTER_SANITIZE_STRING);
3
4 header('Cache-Control: public');
5 header('Content-Type: application/pdf');
6 header('Content-Disposition: inline; filename="' . substr($path, 4) . '"');
7 header('Content-Length: ' . filesize($path));
8 readfile($path);
```

Listing 3.9 – model/displayPDF.php

3.10 model/displayTPIValidation.php

```

1 <?php
2 $tabValidation = '';
3 $tab = '';
4
5 if (in_array('Expert', $_SESSION['roles'][0])) {
6     $tabValidation = getTPIsOfExpert($_SESSION['id']);
7 }
8
9 if (in_array('Manager', $_SESSION['roles'][0])) {
10     $tabValidation = getTPIsOfManager($_SESSION['id']);
11 }
12
13 if (in_array('Administrator', $_SESSION['roles'][0])) {
14     $tabValidation = getTPIsWExpert();
15 }
16 if (empty($tabValidation)) {
17     $assigned = false;
18 }
19
20 foreach ($tabValidation as $value) {
21     if ($value['tpiStatus'] != 'valid') {
22         $tab .= '<tr>';
23         $tab .= '<td>'. $value['tpiID'] . '</td>';
24         $tab .= '<td>'. $value['candidateLastName'] . '</td>';
25         $tab .= '<td>'. $value['candidateFirstName'] . '</td>';
26         $tab .= '<td>'. $value['companyName'] . '</td>';
27         $tab .= '<td>'. $value['managerLastName'] . ' . ' . $value['managerFirstName'] . '</td>';
28         $tab .= '<td>'. $value['sessionStart'] . '</td>';
29         $tab .= '<td>'. $value['sessionEnd'] . '</td>';
30         $tab .= '<td>'. $value['title'] . '</td>';
31         $tab .= '<td>'. $value['cfcDomain'] . '</td>';
32         $tab .= '<td>'. $value['tpiStatus'] . '</td>';
33         $tab .= '<td>'. $value['expert1LastName'] . ' . ' . $value['expert1FirstName'] . '</td>';
34         $tab .= '<td>'. $value['expert2LastName'] . ' . ' . $value['expert2FirstName'] . '</td>';
35         if (!in_array('Candidate', $_SESSION['roles'][0])){
36             if ($value['userExpert1ID'] == $_SESSION['id']) {
37                 $tab .= '<td><a href="?action=checkValidation&tpiID=' . $value['tpiID'] . '&expert=1"&><
button class="btn btn-success">Voir validation </button></a></td>';
38             } elseif ($value['userExpert2ID'] == $_SESSION['id']){
39                 $tab .= '<td><a href="?action=checkValidation&tpiID=' . $value['tpiID'] . '&expert=2"&><
button class="btn btn-success">Voir validation </button></a></td>';
40             } else{
41                 $tab .= '<td><a href="?action=checkValidation&tpiID=' . $value['tpiID'] . '"><button class
="btn btn-success">Voir validation </button></a></td>';
42             }
43         }
44         $tab .= '</tr>';
45     }
46 }

```

Listing 3.10 – model/displayTPIValidation.php

3.11 model/expertWishes.php

```
1 <?php
2 $tpiChoosen = FILTER_INPUT(INPUT_GET, 'idTPI', FILTER_SANITIZE_STRING);
3 $cancelChoice = FILTER_INPUT(INPUT_GET, 'rm', FILTER_SANITIZE_STRING);
```

Listing 3.11 – model/expertWishes.php

3.12 model/formParams.php

```
1 <?php
2 // @TODO check the format of the date and hour aaaa-mm-jj hh:mm:ss
3 require_once 'model/connectDB.php';
4 require_once 'model/crudParams.php';
5
6 $tabParams = getParams();
7
8 const nbExpertMax = 'NbMaxExpertForOneCandidate';
9 const wishesSessionStart = 'WishesSessionStart';
10 const wishesSessionEnd = 'WishesSessionEnd';
11
12 $nbExpMax = $tabParams[nbExpertMax];
13 $dateStart = $tabParams[wishesSessionStart];
14 $dateEnd = $tabParams[wishesSessionEnd];
15
16 $btn = FILTER_INPUT(INPUT_POST, 'send', FILTER_SANITIZE_STRING);
```

Listing 3.12 – model/formParams.php

3.13 model/login.php

```
1 <?php
2 require_once 'connectDB.php';
3 require_once 'crudUser.php';
4
5 $btn = FILTER_INPUT(INPUT_POST, 'send', FILTER_SANITIZE_STRING);
6 $error = '';
```

Listing 3.13 – model/login.php

3.14 model/logout.php

```
1 <?php
2 // Détruit toutes les variables de session
3 $_SESSION = array();
4
5 // Si vous voulez détruire complètement la session, effacez également
6 // le cookie de session.
7 // Note : cela détruira la session et pas seulement les données de session !
8 if (ini_get("session.use_cookies")) {
9     $params = session_get_cookie_params();
10    setcookie(session_name(), '', time() - 42000,
11        $params["path"], $params["domain"],
12        $params["secure"], $params["httponly"]
13    );
14 }
15
16 // Finalement, on détruit la session.
17 session_destroy();
18 header("Location: ?action=login");
19 exit;
```

Listing 3.14 – model/logout.php

3.15 model/printValidation.php

```

1 <style>
2 * {
3     font-family: Arial, sans-serif
4 }
5
6 td {
7     vertical-align: top;
8     padding: 2mm;
9 }
10
11 th {
12     vertical-align: top;
13     text-align: left;
14     padding: 2mm;
15 }
16
17 .attention {
18     color: red;
19 }
20
21 .cartouche,
22 .cartouche td,
23 .cartouche th {
24     border: 1px solid black;
25     border-collapse: collapse;
26 }
27
28 .lowerline {
29     border-bottom: 1px solid black;
30     border-collapse: collapse;
31 }
32
33 h4 {
34     margin: 0;
35 }
36
37 p {
38     margin-top: 1mm;
39     margin-bottom: 1mm;
40 }
41
42 table.page_footer {
43     width: 100%;
44     border: none;
45     border-top: solid 1px black;
46     padding: 2mm
47 }
48
49 table.page_header {
50     width: 100%;
51     border: none;
52     border-bottom: solid 1px black;
53     padding: 2mm
54 }
55 </style>
56
57 <page backtop="10mm" backbottom="10mm" backleft="10mm" backright="10mm">
58
59
60 <table>
61     <tr>
62         <td width="60"></td>
63         <td width="40">
64             <p>République et canton de Genève<br>Département de l'instruction publique, <br>de la
65             formation et de la jeunesse<br><strong>Office pour l'orientation, <br>la formation professionnelle et
66             continue</strong>
67             </p>
68         </td>
69     </tr>
70 </table>
71
72 <h2 style="text-align:center">Travail pratique individuel (TPI)</h2>
73 <h4 style="text-align:center">Informaticien-ne CFC<br>Validation de l'énoncé (<span class="attention
74 ">sans la présence du candidat</span>)</h4>
75
76 <table class="cartouche">
77     <tr>
78         <td width="310">
79             <h4>Entreprise formatrice, chef de projet</h4>
80             <p>
81                 <?php echo $managerCompagny ?><br>
82                 <?php echo $managerLastName ?> <?php echo $managerFirstName ?><br>
83                 <?php echo $managerPhone ?><br>
84                 <?php echo $managerMail ?>

```

```

83         </p>
84     </td>
85     <td width="310">
86         <h4>Candidat </h4>
87         <p><br>
88             <?php echo $candLastName ?> <?php echo $candFirstName ?><br>
89             <?php echo $candPhone ?><br>
90             <?php echo $candMail ?>
91         </p>
92     </td>
93
94 </tr>
95 <tr>
96     <td>
97         <h4>1<sup>er</sup> Expert </h4>
98         <p>
99             <?php echo $expert1LastName ?> <?php echo $expert1FirstName ?><br>
100             <?php echo $expert1Phone ?><br>
101             <?php echo $expert1Mail ?>
102         </p>
103     </td>
104     <td>
105         <h4>2<sup>ème</sup> Expert </h4>
106         <p>
107             <?php echo $expert2LastName ?> <?php echo $expert2FirstName ?><br>
108             <?php echo $expert2Phone ?><br>
109             <?php echo $expert2Mail ?>
110         </p>
111     </td>
112 </tr>
113 </table>
114 <p><br></p>
115
116 <table>
117     <tr>
118         <th width="100" class="lowerline">Titre du travail </th>
119         <td width="510" class="lowerline">
120             <h4><?php echo $title ?></h4>
121         </td>
122     </tr>
123
124     <tr>
125         <th width="100" class="lowerline">Domaine </th>
126         <td width="510" class="lowerline"><?php echo $domain ?></td>
127     </tr>
128
129     <tr>
130         <th width="100" class="lowerline">Dates </th>
131         <td width="510" class="lowerline">du <?php echo $start ?> au <?php echo $end ?>, de <?php echo
132             $hourStart ?> à <?php echo $hourEnd ?></td>
133     </tr>
134
135     <tr>
136         <th width="100" class="lowerline">Lieu où se déroule le travail </th>
137         <td width="510" class="lowerline"><?php echo $workplace ?></td>
138     </tr>
139 </table>
140
141 <table class="cartouche">
142     <thead>
143         <tr>
144             <th width="520">Critère </th>
145             <th width="100">Rempli </th>
146         </tr>
147     </thead>
148     <tbody>
149 <?php
150 foreach ($validation_criteria as $key => $value) {
151     <tr>
152         <td width="500"> <?php echo $value ?> </td>
153         <td width="100"> <?php echo $tabNewCrit[$key] ?> </td>
154     </tr>
155 <?php } ?>
156 <tr>
157     <td colspan="2" width="640">
158         <h4>Remarques, mesures de correction </h4>
159         <p><?php echo $comment ?></p>
160     </td>
161 </tr>
162 </tbody>
163 </table>
164 <h2>Validation </h2>
165
166 <table class="cartouche">
167     <tbody>
168     <tr>
169         <td width="640">
170             <?php echo $expert1Valid ?>

```

```
170 |         <br>
171 |         <?php echo $expert2Valid ?>
172 |     </td>
173 | </tr>
174 | </tbody>
175 | </table>
176 | </page>
```

Listing 3.15 – model/printValidation.php

3.16 model/selectExpert.php

```

1 <?php
2 $tpiChoosen = FILTER_INPUT(INPUT_GET, 'idTPI', FILTER_SANITIZE_STRING);
3 $idExpert = FILTER_INPUT(INPUT_GET, 'idExpert', FILTER_SANITIZE_STRING);
4 $assigned = FILTER_INPUT(INPUT_GET, 'assigned', FILTER_SANITIZE_STRING);
5
6 $tab = "";
7
8 $nbExpert = 0;
9 foreach (getTPIsById($tpiChoosen) as $value) {
10     $tab .= '<td>' . $value['tpiID'] . '</td>';
11     $tab .= '<td>' . $value['candidateLastName'] . '</td>';
12     $tab .= '<td>' . $value['candidateFirstName'] . '</td>';
13     $tab .= '<td>' . $value['companyName'] . '</td>';
14     $tab .= '<td>' . $value['managerLastName'] . '</td>' . $value['managerFirstName'] . '</td>';
15     $tab .= '<td>' . $value['sessionStart'] . '</td>';
16     $tab .= '<td>' . $value['sessionEnd'] . '</td>';
17     $tab .= '<td><a href="pdf/Enonce_TPI_' . $value['year'] . '_' . $value['tpiID'] . '_' . $value['candidateLastName'] . '_' . $value['candidateFirstName'] . '.pdf">' . $value['title'] . '</a></td>';
18     $tab .= '<td>' . $value['cfcDomain'] . '</td>';
19     $tab .= '<td>';
20     $names = getWishesByTpilDAssignedNull($value['tpiID']);
21     $nbExpertAssigned = count(getWishesByTpilD($value['tpiID']));
22     foreach ($names as $key => $name) {
23         $key++;
24         $tab .= '<div class="row">';
25         $tab .= '<div class="col-5">' . $key . ' . ' . $name['expertLastName'] . ' . ' . $name['expertFirstName'] . '</div>';
26         if ($nbExpertAssigned < 2) {
27             $tab .= '<a href="?action=chooseExpert&idTPI=' . $value['tpiID'] . '&idExpert=' . $name['userExpertID'] . '&assigned=1"><button class="ml-2 mb-2 btn btn-outline-success">Expert 1</button></a>';
28             $tab .= '<a href="?action=chooseExpert&idTPI=' . $value['tpiID'] . '&idExpert=' . $name['userExpertID'] . '&assigned=2"><button class="ml-2 mb-2 btn btn-outline-success">Expert 2</button></a></div><br>';
29         } else {
30             $tab .= '<a><button class="ml-2 mb-2 btn btn-secondary" disabled>Expert 1</button></a>';
31             $tab .= '<a><button class="ml-2 mb-2 btn btn-secondary" disabled>Expert 2</button></a></div><br>';
32         }
33     }
34     $tab .= '</td></tr>';
35 }

```

Listing 3.16 – model/selectExpert.php

3.17 model/tabTPI.php

```

1 <?php
2 require_once 'connectDB.php';
3 require_once 'crudWishes.php';
4 require_once 'crudParams.php';
5
6 session_start();
7
8 $order = filter_input(INPUT_GET, 'order', FILTER_SANITIZE_STRING);
9 $order = utf8_encode($order);
10 $order = str_replace('&#34;', '"', $order);
11 $order = str_replace('\\', '\\', $order);
12 $tblorder = json_decode($order);
13
14 $limit = filter_input(INPUT_GET, 'limit', FILTER_SANITIZE_NUMBER_INT);
15 $page = filter_input(INPUT_GET, 'page', FILTER_SANITIZE_NUMBER_INT);
16
17 $filter = filter_input(INPUT_GET, 'filter');
18 $filter = utf8_encode($filter);
19 $filter = str_replace('&#34;', '"', $filter);
20 $filter = str_replace('\\', '\\', $filter);
21 $tblfilter = json_decode($filter, true);
22
23 $query = "SELECT tpiID, pdfPath, year, tpiStatus, title, cfcDomain, sessionStart, sessionEnd,
presentationDate, workplace, userCandidateID, userManagerID, userExpert1ID, uc.LastName AS
candidateLastName, uc.FirstName AS candidateFirstName, um.LastName AS managerLastName, um.FirstName AS
managerFirstName, ue1.LastName AS expert1LastName, ue1.FirstName AS expert1FirstName, ue2.LastName AS
expert2LastName, ue2.FirstName AS expert2FirstName, tpiStatus, submissionDate, uc.companyName FROM
tpis LEFT JOIN users AS uc ON userCandidateID = uc.userID LEFT JOIN users AS um ON userManagerID = um.
userID LEFT JOIN users AS ue1 ON userExpert1ID = ue1.userID LEFT JOIN users AS ue2 ON userExpert2ID =
ue2.userID";
24
25 $sqlField = ['tpiID', 'pdfPath', 'uc.LastName', 'uc.FirstName', 'um.LastName', 'um.FirstName', '
companyName', 'ue1.LastName', 'ue1.FirstName', 'ue2.LastName', 'ue2.FirstName', 'sessionStart', '
sessionEnd', 'title', 'cfcDomain', 'tpiStatus'];
26
27 $sqlField2 = ['tpiID', 'pdfPath', 'candidateLastName', 'candidateFirstName', 'managerLastName', '
managerFirstName', 'companyName', 'expert1LastName', 'expert1FirstName', 'expert2LastName', '
expert2FirstName', 'sessionStart', 'sessionEnd', 'title', 'cfcDomain', 'tpiStatus'];
28
29 $conditionOperator = ['=', '<=', '<', '>', '>=', 'like'];
30
31 $conditionValue = [];
32
33 if (!empty($tblfilter)) {
34     $query .= " WHERE ";
35     foreach ($tblfilter as $key => $value) {
36         if (in_array($key, $sqlField)) {
37             if (array_key_first($tblfilter) == $key) {
38                 $query .= " (";
39             } else {
40                 $query .= " AND (";
41             }
42             foreach ($value as $condition) {
43                 $count = count($conditionValue);
44                 if (in_array($condition[0], $conditionOperator)) {
45                     if ($condition[0] == 'like') {
46                         array_push($conditionValue, '%' . $condition[1] . '%');
47                     } else {
48                         array_push($conditionValue, $condition[1]);
49                     }
50                     if ($value[0] == $condition) {
51                         $query .= $key . ' ' . $condition[0] . ' ' . ':param' . $count;
52                     } else {
53                         $query .= ' OR ' . $key . ' ' . $condition[0] . ' ' . ':param' . $count;
54                     }
55                 }
56             }
57             $query .= ")";
58         }
59     }
60 }
61 }
62
63 if (!empty($tblorder)) {
64     foreach ($tblorder as $key => $col) {
65         if (($col[1] == 'ASC' || $col[1] == 'DESC') && in_array($col[0], $sqlField2)) {
66             if (array_key_first($tblorder) == $key) {
67                 $query .= " ORDER BY " . $col[0] . " " . $col[1];
68             } else {
69                 $query .= ", " . $col[0] . " " . $col[1];
70             }
71         }
72     }
73 }

```



```

73 }
74 }
75
76 $order = getConnexion();
77 $req = $order->prepare($query);
78
79 for ($i = 0; $i < count($conditionValue); $i++) {
80     $req->bindParam(":param" . $i, $conditionValue[$i], PDO::PARAM_STR);
81 }
82
83 $req->execute();
84 $res = $req->fetchAll(PDO::FETCH_ASSOC);
85 //Display the table in the page tabTPI
86 foreach ($res as $key => $value) {
87     echo '<tr>';
88     echo '<th scope="row">' . $value['tpiID'] . '</th>';
89     echo '<td>' . $value['candidateLastName'] . '</td>';
90     echo '<td>' . $value['candidateFirstName'] . '</td>';
91     echo '<td>' . $value['companyName'] . '</td>';
92     echo '<td>' . $value['managerLastName'] . '</td>' . $value['managerFirstName'] . '</td>';
93     echo '<td>' . $value['sessionStart'] . '</td>';
94     echo '<td>' . $value['sessionEnd'] . '</td>';
95     echo '<td><a target="blank" href="?action=displayPDF&link=pdf/' . $value['pdfPath'] . '>' . $value['
title'] . '</a></td>';
96     echo '<td>' . $value['cfcDomain'] . '</td>';
97     echo '<td>' . $value['tpiStatus'] . '</td>';
98     echo '<td>' . $value['expert1LastName'] . '</td>' . $value['expert1FirstName'] . '</td>';
99     echo '<td>' . $value['expert2LastName'] . '</td>' . $value['expert2FirstName'] . '</td>';
100     echo '<td>';
101     $names = getWishesByTpilDAssignedNull($value['tpiID']);
102     $nbExpert = count($names);
103     foreach ($names as $key => $name) {
104         $key++;
105         echo $key . ' . ' . $name['expertLastName'] . ' . ' . $name['expertFirstName'] . '<br>';
106     }
107     echo '</td>';
108
109     // If the number of expert choices is inferior of the number max of expert per TPI (in table params)
110     if ($nbExpert < getParamsByName('NbMaxExpertForOneCandidate')[0]['value']) {
111         echo '<td>';
112         if (in_array('Expert', $_SESSION['roles'][0])) {
113             //check if the date is between the date define by the admin
114             if (date('Y-m-d H:i:m') >= getParamsByName('WishesSessionStart')[0]['value'] && date('Y-m-d H:
i:m') <= getParamsByName('WishesSessionEnd')[0]['value']) {
115                 if (getWishUser($_SESSION['id'], $value['tpiID'])) {
116                     echo '<a href="?action=displayTPI&idTPI=' . $value['tpiID'] . '&rm=true"><button class
="btn btn-danger">Annuler</button></a>';
117                 } else {
118                     if (empty($value['expert1LastName']) || empty($value['expert2LastName'])) {
119                         echo '<a href="?action=displayTPI&idTPI=' . $value['tpiID'] . '><button class="
btn btn-success">Choisir</button></a>';
120                     } else {
121                         echo '<a><button class="btn btn-secondary" disabled>Choisir</button></a>';
122                     }
123                 }
124             } else {
125                 echo '<a><button class="btn btn-secondary" disabled>Session fermée</button></a>';
126             }
127         }
128
129         if (in_array('Administrator', $_SESSION['roles'][0])) {
130             if (empty($value['expert1LastName']) || empty($value['expert2LastName'])) {
131                 echo '<a href="?action=chooseExpert&idTPI=' . $value['tpiID'] . '><button class="btn btn-
success">Choisir Expert</button></a>';
132             } else {
133                 echo '<a><button class="btn btn-secondary" disabled>Choisir Expert</button></a>';
134             }
135         }
136     }
137     echo '</td>';
138 } else {
139     echo '<td>';
140     if (in_array('Expert', $_SESSION['roles'][0])) {
141         if (date('Y-m-d H:i:m') >= getParamsByName('WishesSessionStart')[0]['value'] && date('Y-m-d H:
i:m') <= getParamsByName('WishesSessionEnd')[0]['value']) {
142             if (getWishUser($_SESSION['id'], $value['tpiID'])) {
143                 echo '<a href="?action=displayTPI&idTPI=' . $value['tpiID'] . '&rm=true"><button class
="btn btn-danger">Annuler</button></a>';
144             } else {
145                 echo '<button class="btn btn-secondary" disabled>Choisir</button>';
146             }
147         } else {
148             echo '<a><button class="btn btn-secondary" disabled>Session fermée</button></a>';
149         }
150     }
151     if (in_array('Administrator', $_SESSION['roles'][0])) {

```

```
153 |         echo ' <a href="?action=chooseExpert&idTPI=' . $value['tpiID'] . '"><button class="btn btn-  
154 |             secondary">Choisir Expert </button></a>';  
155 |     }  
156 |     echo ' </td>';  
157 | }  
158 | echo ' </tr>';  
159 | }
```

Listing 3.17 – model/tabTPI.php

Chapitre 4

Vues

4.1 view/editValidation.php

```
1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5     <meta charset="utf-8" />
6     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8     <meta name="description" content="" />
9     <meta name="author" content="" />
10    <title>Collège d'Experts Informatique de Genève</title>
11    <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12         sb-admin/) -->
13    <link href="css/styles.css" rel="stylesheet" />
14    <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15         crossorigin="anonymous" />
16    <script src="https://cdn.jsdelivr.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17         anonymous"></script>
18 </head>
19
20 <body class="sb-nav-fixed">
21     <?php require_once 'navH.php'; ?>
22     <div id="layoutSidenav">
23         <?php require_once 'navV.php'; ?>
24         <div id="layoutSidenav_content">
25             <main>
26                 <!-- Début du contenu de la page -->
27                 <div class="container-fluid">
28                     <?php $form ?>
29                 </div>
30                 <!-- fin du contenu de la page -->
31             </main>
32             <?php require_once 'footer.php'; ?>
33         </div>
34     </div>
35     <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
36     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
37         crossorigin="anonymous"></script>
38     <script src="js/scripts.js"></script>
39     <script src="https://cdn.jsdelivr.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
40         anonymous"></script>
41 </body>
42 </html>
```

Listing 4.1 – view/editValidation.php

4.2 view/footer.php

```
1 <footer class="py-4 bg-light mt-auto">
2   <div class="container-fluid">
3     <div class="d-flex align-items-center justify-content-between small">
4       <div class="text-muted">TPI 2020 – développé par NGUYEN KELLY</div>
5       <div class="text-muted">Application en cours de développement</div>
6     </div>
7   </div>
8 </footer>
```

Listing 4.2 – view/footer.php

4.3 view/formParams.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8   <meta name="description" content="" />
9   <meta name="author" content="" />
10  <title>Collège d'Experts Informatique de Genève</title>
11  <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12  sb-admin/) -->
13  <link href="css/styles.css" rel="stylesheet" />
14  <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15  crossorigin="anonymous" />
16  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17  anonymous"></script>
18 </head>
19
20 <body class="sb-nav-fixed">
21   <?php require_once 'navH.php'; ?>
22   <div id="layoutSidenav">
23     <?php require_once 'navV.php'; ?>
24     <div id="layoutSidenav_content">
25       <main>
26         <!-- Début du contenu de la page -->
27         <div class="container-fluid">
28           <form class="mt-4" action="?action=editParam" method="POST">
29             <div class="row">
30               <div class="form-group col-4">
31                 <label for="dateStart">Date de début</label>
32                 <input pattern="^\d{4}(-)((0)[0-9]|((1)[0-2]))(-)[0-2][0-9]((3)[0-1])((
33                 )([0-2][0-9])|(:)([0-6][0-9])|(:)([0-6][0-9])$)" class="form-control" type="text" id="dateStart" name="
34                 dateStart" value="<?= $dateStart ?>">
35                 <small id="dateStart" class="form-text text-muted">Format : aaaa-mm-jj hh:
36                 mm:ss</small>
37               </div>
38               <div class="form-group col-4">
39                 <label for="dateEnd">Date de fin</label>
40                 <input pattern="^\d{4}(-)((0)[0-9]|((1)[0-2]))(-)[0-2][0-9]((3)[0-1])((
41                 )([0-2][0-9])|(:)([0-6][0-9])|(:)([0-6][0-9])$)" class="form-control" type="text" id="dateEnd" name="
42                 dateEnd" value="<?= $dateEnd ?>">
43                 <small id="dateEnd" class="form-text text-muted">Format : aaaa-mm-jj hh:mm
44                 :ss</small>
45               </div>
46               <div class="form-group col-4">
47                 <label for="nbExpMax">Nombre d'expert maximum par TPI</label>
48                 <input class="form-control" type="number" id="nbExpMax" name="nbExpMax"
49                 value="<?= $nbExpMax ?>">
50               </div>
51             </div>
52             <button type="submit" name="send" value="send" class="btn btn-success">Enregistrer </
53             button>
54           </form>
55           <!-- fin du contenu de la page -->
56         </div>
57       </main>
58     <?php require_once 'footer.php'; ?>
59   </div>
60 </div>
61
62 <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
63 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
64  crossorigin="anonymous"></script>
65 <script src="js/scripts.js"></script>
66 <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
67  anonymous"></script>
68 </body>
69 </html>

```

Listing 4.3 – view/formParams.php

4.4 view/home.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5     <meta charset="utf-8" />
6     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8     <meta name="description" content="" />
9     <meta name="author" content="" />
10    <title>Collège d'Experts Informatique de Genève</title>
11    <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12         sb-admin/) -->
13    <link href="css/styles.css" rel="stylesheet" />
14    <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15         crossorigin="anonymous" />
16    <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17         anonymous"></script>
18 </head>
19
20 <body class="sb-nav-fixed">
21     <?php require_once 'navH.php'; ?>
22     <div id="layoutSidenav">
23         <?php require_once 'navV.php'; ?>
24         <div id="layoutSidenav_content">
25             <main>
26                 <!-- Début du contenu de la page -->
27                 <div class="container-fluid">
28                     <h1 class="mt-4">Collège d'Experts Informatique de Genève</h1>
29                     <h2>Outil de collaboration et de gestion des TPIs </h2>
30                     <p>Le but de cette application est de donner aux membres du collège d'experts en
31                         informatique
32                         du canton de Genève un outil leur permettant de gérer leur travail tout au long de
33                         l'année scolaire.
34                         Le développement a été confié a des élèves de l'école d'informatique, dans le
35                         cadre de leur TPI.
36                         Les modules développés initialement sont :</p>
37                         <ul>
38                             <li>L'administration des utilisateurs</li>
39                             <li>La rédaction des énoncés des TPIs</li>
40                             <li>La répartition des TPIs entre les experts</li>
41                             <li>La validation des énoncés</li>
42                             <li>L'évaluation des TPis</li>
43                         </ul>
44                         <p>Cette partie du développement concerne le module de répartition et validation,
45                         développé par NGUYEN KELLY</p>
46                     </div>
47                 <!-- fin du contenu de la page-->
48             </main>
49             <?php require_once 'footer.php'; ?>
50         </div>
51     </div>
52     <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
53     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
54         crossorigin="anonymous"></script>
55     <script src="js/scripts.js"></script>
56     <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
57         anonymous"></script>
58 </body>
59
60 </html>

```

Listing 4.4 – view/home.php

4.5 view/login.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="utf-8" />
5   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
7   <meta name="description" content="" />
8   <meta name="author" content="" />
9   <title>Collège d'Experts Informatique de Genève</title>
10  <link href="css/styles.css" rel="stylesheet" />
11  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
12  anonymous"></script>
13 </head>
14 <body class="bg-light">
15   <div id="layoutAuthentication">
16     <div id="layoutAuthentication_content">
17       <main>
18         <div class="container">
19           <div class="row justify-content-center">
20             <div class="col-lg-5">
21               <div class="card shadow-lg border-0 rounded-lg mt-5">
22                 <div class="card-header"><h3 class="text-center font-weight-light my-4"><
23                 img src="./img/LogoExpertsDev.svg" width="60" height="60" class="d-inline-block align-middle mr-5" alt
24                 =" " loading="lazy">CdEIG - Identification </h3></div>
25                 <div class="card-body">
26                   <form action="?action=login" method="POST">
27                     <div class="form-group">
28                       <label class="small mb-1" for="inputEmailAddress">Email</label>
29
30                     <input class="form-control py-4" id="inputEmailAddress" name="
31                     inputEmailAddress" type="email" placeholder="Saisissez votre email" value="<?= empty($email) ? ' ' :
32                     $email ?">"/>
33
34                     </div>
35                     <div class="form-group">
36                       <label class="small mb-1" for="inputPassword">Mot de passe</
37                       label>
38
39                       <input class="form-control py-4" id="inputPassword" name="
40                       inputPassword" type="password" placeholder="Saisissez votre mot de passe" />
41
42                       <?php
43                         if ($error) {
44                           echo '<small class="form-text text-danger">Vos
45                           identifiants sont incorrects </small>';
46                         }
47                       <?>
48                     </div>
49                     <div class="form-group d-flex align-items-center justify-content-
50                     between mt-4 mb-0"><button class="btn btn-primary" name="send" type="submit" value="send">Login </
51                     button></div>
52                   </form>
53                 </div>
54                 <div class="card-footer text-center">
55                   <div>
56                     </div>
57                 </div>
58             </div>
59           </div>
60         </div>
61       </main>
62     </div>
63     <div id="layoutAuthentication_footer">
64       <?php require_once 'footer.php'; ?>
65     </div>
66   </div>
67   <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
68   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
69   crossorigin="anonymous"></script>
70   <script src="js/scripts.js"></script>
71 </body>
72 </html>

```

Listing 4.5 – view/login.php

4.6 view/navH.php

```

1 <nav class="sb-topnav navbar navbar-expand navbar-light" style="background-color:lightgray">
2   <a class="navbar-brand" href="#">
3     
5     CdEIG
6   </a>
7   <button class="btn btn-link btn-sm order-1 order-lg-0" id="sidebarToggle" href="#"><i class="fas fa-
8     bars"></i></button>
9   <!-- Navbar Search-->
10  <div class="d-none d-md-inline-block ml-auto mr-0 mr-md-3 my-2 my-md-0">
11    <ul class="navbar-nav mr-auto">
12      <!-- Menu principal quand l'utilisateur est identifié -->
13      <li class="nav-item active">
14        <a class="nav-link" href="#">A propos <span class="sr-only">(current) </span></a>
15      </li>
16      <li class="nav-item">
17        <a class="nav-link" href="?action=selectTPI">Répartition </a>
18      </li>
19      <li class="nav-item">
20        <a class="nav-link" href="#">Validation </a>
21      </li>
22      <li class="nav-item">
23        <a class="nav-link" href="#">Module 3</a>
24      </li>
25      <li class="nav-item">
26        <a class="nav-link" href="#">Module 4</a>
27      </li>
28      <!-- fin du menu principal -->
29    </ul>
30  </div>
31  <!-- Navbar-->
32  <ul class="navbar-nav ml-auto ml-md-0">
33    <li class="nav-item dropdown">
34      <a class="nav-link dropdown-toggle" id="userDropdown" href="#" role="button" data-toggle="
35        dropdown" aria-haspopup="true" aria-expanded="false"><i class="fas fa-user fa-fw"></i></a>
36      <div class="dropdown-menu dropdown-menu-right" aria-labelledby="userDropdown">
37        <a class="dropdown-item" href="#">Paramètres </a>
38        <div class="dropdown-divider"></div>
39        <a class="dropdown-item" href="?action=logout">Déconnexion </a>
40      </div>
41    </li>
42  </ul>
</nav>

```

Listing 4.6 – view/navH.php

4.7 view/navV.php

```

1 <?php
2
3 $pagesName_distribution_byRight = [
4     "displayTPI" => "Tableau TPI",
5     "editParam" => "Formulaire Admin",
6 ];
7
8 $pagesName_validation_byRight = [
9     "displayValidationTPI" => "Afficher les Validations"
10 ];
11
12 ?>
13
14 <div id="layoutSidenav_nav">
15     <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
16         <div class="sb-sidenav-menu">
17             <div class="nav">
18                 <!-- Menu secondaire (qui reprend le menu principal...) -->
19                 <a class="nav-link" href="home.php">A propos</a>
20
21                 <div class="sb-sidenav-menu-heading">Répartition TPI</div>
22                 <?php
23                 foreach ($pagesName_distribution_byRight as $key => $value) {
24                     if (in_array($key, $_SESSION['rights'][0])) {
25                         echo '<a class="nav-link" href="?action=' . $key . '">' . $value . '</a>';
26                     }
27                 }
28                 ?>
29                 <div class="sb-sidenav-menu-heading">Validation</div>
30                 <?php
31                 foreach ($pagesName_validation_byRight as $key => $value) {
32                     if (in_array($key, $_SESSION['rights'][0])) {
33                         echo '<a class="nav-link" href="?action=' . $key . '">' . $value . '</a>';
34                     }
35                 }
36                 ?>
37                 <div class="sb-sidenav-menu-heading">Module 3</div>
38                 <a class="nav-link" href="home.php">Page/ Action 1</a>
39                 <a class="nav-link" href="home.php">Page/ Action 2</a>
40                 <a class="nav-link" href="home.php">Page/ Action 3</a>
41                 <a class="nav-link" href="home.php">Page/ Action 4</a>
42                 <div class="sb-sidenav-menu-heading">Module 4</div>
43                 <a class="nav-link" href="home.php">Page/ Action 1</a>
44                 <a class="nav-link" href="home.php">Page/ Action 2</a>
45                 <a class="nav-link" href="home.php">Page/ Action 3</a>
46                 <a class="nav-link" href="home.php">Page/ Action 4</a>
47                 <!-- Fin du menu secondaire -->
48             </div>
49         </div>
50         <div class="sb-sidenav-footer">
51             <div class="small">Connecté en tant que:</div>
52             <?= $_SESSION['name'] ?> <br>
53             <?php foreach($_SESSION["roles"][0] as $r) echo $r . ' '; ?>
54         </div>
55     </nav>
56 </div>

```

Listing 4.7 – view/navV.php

4.8 view/selectExpert.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8   <meta name="description" content="" />
9   <meta name="author" content="" />
10  <title>Collège d'Experts Informatique de Genève</title>
11  <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12  sb-admin/) -->
13  <link href="css/styles.css" rel="stylesheet" />
14  <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15  crossorigin="anonymous" />
16  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17  anonymous"></script>
18 </head>
19
20 <body class="sb-nav-fixed">
21   <?php require_once 'navH.php'; ?>
22   <div id="layoutSidenav">
23     <?php require_once 'navV.php'; ?>
24     <div id="layoutSidenav_content">
25       <main>
26         <!-- Début du contenu de la page -->
27         <div class="container-fluid">
28           <table class="table">
29             <thead>
30               <tr>
31                 <th scope="col">#</th>
32                 <th scope="col">Nom</th>
33                 <th scope="col">Prénom</th>
34                 <th scope="col">Entreprise</th>
35                 <th scope="col">Chef de projet</th>
36                 <th scope="col">Date de début</th>
37                 <th scope="col">Date de fin</th>
38                 <th scope="col">Titre</th>
39                 <th scope="col">Domaine</th>
40                 <th scope="col">Souhait Experts</th>
41               </tr>
42             </thead>
43             <tbody>
44               <?php $tab = $tab ?>
45             </tbody>
46           </table>
47           <a href="?action=displayTPI"><button class="btn btn-primary float-right">Retour</
48           button></a>
49         </div>
50         <!-- fin du contenu de la page -->
51       </main>
52       <?php require_once 'footer.php'; ?>
53     </div>
54   </div>
55   <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
56   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
57   crossorigin="anonymous"></script>
58   <script src="js/scripts.js"></script>
59   <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
60   anonymous"></script>
61 </body>
62 </html>

```

Listing 4.8 – view/selectExpert.php

4.9 view/tabTPI.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8   <meta name="description" content="" />
9   <meta name="author" content="" />
10  <title>Collège d'Experts Informatique de Genève</title>
11  <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12  sb-admin/) -->
13  <link href="css/styles.css" rel="stylesheet" />
14  <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15  crossorigin="anonymous" />
16  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17  anonymous"></script>
18  <style>
19    th[onclick] {
20      cursor: pointer;
21    }
22
23    #model {
24      visibility: hidden;
25    }
26
27    #colNameModel {
28      visibility: hidden;
29    }
30
31    .cancel {
32      width: 20px;
33      height: 20px;
34    }
35  </style>
36 </head>
37
38 <body class="sb-nav-fixed">
39   <?php require_once 'navH.php'; ?>
40   <div id="layoutSidenav">
41     <?php require_once 'navV.php'; ?>
42     <div id="layoutSidenav_content">
43       <main>
44         <!-- Début du contenu de la page -->
45         <div class="container-fluid">
46           <div class="alert alert-primary mt-2">
47             Info
48             <br>simple clique : tri selon la colonne
49             <br>shift + clique : inverse l'ordre de tri (ou ajoute en ascendant si absent du tri)
50             <br>ctrl + clique : ajoute ou supprime une colonne du tri
51           </div>
52           <div id="form"></div>
53           <button class="btn btn-outline-success mt-2 mb-2" onclick="addFilter()">Ajouter une
54           condition</button>
55           <table class="table">
56             <thead>
57               <tr id="colName">
58                 <th scope="col">#</th>
59                 <th id="candidateLastName" onclick="setOrder(this)" scope="col">Nom</th>
60                 <th id="candidateFirstName" onclick="setOrder(this)" scope="col">Prénom</th>
61                 <th id="companyName" onclick="setOrder(this)" scope="col">Entreprise</th>
62                 <th id="managerLastName" onclick="setOrder(this)" scope="col">Chef de projet</th>
63               </tr>
64             </thead>
65             <tbody id="dataContainer"></tbody>
66           </table>
67           <table id="colNameModel">
68             <thead>
69               <tr id="colName">
70                 <th scope="col">#</th>
71                 <th id="candidateLastName" onclick="setOrder(this)" scope="col">Nom</th>
72                 <th id="candidateFirstName" onclick="setOrder(this)" scope="col">Prénom</th>
73                 <th id="companyName" onclick="setOrder(this)" scope="col">Entreprise</th>
74                 <th id="managerLastName" onclick="setOrder(this)" scope="col">Chef de projet</th>
75               </tr>
76             </thead>
77             <tbody id="dataContainer"></tbody>
78           </table>
79         </div>
80       </main>
81     </div>
82   </div>
83 </body>
84 </html>

```

```

80         <th id="sessionStart" onclick="setOrder(this)" scope="col">Date de début</th>
81         <th id="sessionEnd" onclick="setOrder(this)" scope="col">Date de fin</th>
82         <th id="title" onclick="setOrder(this)" scope="col">Titre</th>
83         <th id="cfcDomain" onclick="setOrder(this)" scope="col">Domaine</th>
84         <th id="tpiStatus" onclick="setOrder(this)" scope="col">Status</th>
85         <th id="expert1LastName" onclick="setOrder(this)" scope="col">Expert 1</th>
86         <th id="expert2LastName" onclick="setOrder(this)" scope="col">Expert 2</th>
87         <th scope="col">Souhait Experts</th>
88         <th scope="col">Choix</th>
89     </tr>
90 </thead>
91 </table>
92 <div class="form-inline mt-1" id="model">
93     <select class="form-control" onchange="updateFilter()">
94         <option value="uc.LastName">Nom</option>
95         <option value="uc.FirstName">Prénom</option>
96         <option value="companyName">Entreprise</option>
97         <option value="um.LastName">Chef de projet</option>
98         <option value="sessionStart">Date de début</option>
99         <option value="sessionEnd">Date de fin</option>
100         <option value="title">Titre</option>
101         <option value="cfcDomain">Domaine</option>
102         <option value="ue1.LastName">Expert 1</option>
103         <option value="ue2.LastName">Expert 2</option>
104     </select>
105     <select class="ml-1 form-control" onchange="updateFilter()">
106         <option><=</option>
107         <option><=</option>
108         <option><<=</option>
109         <option>>></option>
110         <option>>></option>
111         <option value="like">comme</option>
112     </select>
113     <input type="text" class="ml-1 form-control" onchange="updateFilter()" onkeypress="
updateFilter()" onkeydown="updateFilter()" onkeyup="updateFilter()">
114     <button class="ml-1 btn btn-outline-danger" onclick="removeFilter(this.parentNode)">
Supprimer</button>
115 </div>
116 </div>
117 <!-- fin du contenu de la page-->
118 </main>
119 <?php require_once 'footer.php' ?>
120 </div>
121 <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
122 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
123 <script src="js/scripts.js"></script>
124 <script src="js/tabOrder.js"></script>
125 <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
anonymous"></script>
126 </body>
127
128 </html>

```

Listing 4.9 – view/tabTPI.php

4.10 view/validateTPIs.php

```

1 <!DOCTYPE html>
2 <html lang="fr">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
8   <meta name="description" content="" />
9   <meta name="author" content="" />
10  <title>Collège d'Experts Informatique de Genève</title>
11  <!-- Mise en page basée sur le template Start Bootstrap - Admin (https://startbootstrap.com/templates/
12  sb-admin/) -->
13  <link href="css/styles.css" rel="stylesheet" />
14  <link href="https://cdn.datatables.net/1.10.20/css/dataTables.bootstrap4.min.css" rel="stylesheet"
15  crossorigin="anonymous" />
16  <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/js/all.min.js" crossorigin="
17  anonymous"></script>
18 </head>
19 <body class="sb-nav-fixed">
20   <?php require_once 'navH.php'; ?>
21   <div id="layoutSidenav">
22     <?php require_once 'navV.php'; ?>
23     <div id="layoutSidenav_content">
24       <main>
25         <!-- Début du contenu de la page -->
26         <div class="container-fluid">
27           <table class="table">
28             <thead>
29               <tr>
30                 <th scope="col">#</th>
31                 <th scope="col">Nom</th>
32                 <th scope="col">Prénom</th>
33                 <th scope="col">Entreprise</th>
34                 <th scope="col">Chef de projet</th>
35                 <th scope="col">Date de début</th>
36                 <th scope="col">Date de fin</th>
37                 <th scope="col">Titre</th>
38                 <th scope="col">Domaine</th>
39                 <th scope="col">Statut</th>
40                 <th scope="col">Expert 1</th>
41                 <th scope="col">Expert 2</th>
42                 <th scope="col">Validation</th>
43             </tr>
44             </thead>
45             <tbody>
46               <?php $tab = $table ?>
47               <?php $tab = $table ?>
48             </tbody>
49           </table>
50           <div class="mx-auto" style="width: 400px;">
51             <?php $assigned = $table ?>
52             <h3 class="text-danger">Vous êtes assigné à aucun TPI</h3> : ''
53           </div>
54         </div>
55         <!-- fin du contenu de la page -->
56       </main>
57       <?php require_once 'footer.php'; ?>
58     </div>
59   </div>
60   <script src="https://code.jquery.com/jquery-3.4.1.min.js" crossorigin="anonymous"></script>
61   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"
62   crossorigin="anonymous"></script>
63   <script src="js/scripts.js"></script>
64   <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js" crossorigin="
65   anonymous"></script>
66 </body>
67 </html>

```

Listing 4.10 – view/validateTPIs.php

Chapitre 5

Controlleurs

5.1 controller/checkValidation.php

```
1 <?php
2 require_once 'model/crudTPIS.php';
3 require_once 'model/crudValidation.php';
4 require_once 'model/checkValidation.php';
5 require_once 'view/editValidation.php';
```

Listing 5.1 — controller/checkValidation.php

5.2 controller/chooseExpert.php

```
1 <?php
2 require_once 'model/crudWishes.php';
3 require_once 'model/crudTPIs.php';
4 require_once 'model/selectExpert.php';
5
6 if (is_numeric($idExpert) && is_numeric($tpiChoosen) && is_numeric($assigned)) {
7     updWish($idExpert, $tpiChoosen, $assigned);
8     if ($assigned == '1') {
9         updTPIExp1($tpiChoosen, $idExpert);
10    } else {
11        updTPIExp2($tpiChoosen, $idExpert);
12    }
13 }
14
15 require_once 'view/selectExpert.php';
```

Listing 5.2 – controller/chooseExpert.php

5.3 controller/displayPDF.php

```
1 <?php
2 require_once 'model/displayPDF.php';
```

Listing 5.3 – controller/displayPDF.php

5.4 controller/displayTPI.php

```
1 <?php
2 require_once 'model/crudParams.php';
3 require_once 'model/crudWishes.php';
4 require_once 'model/crudTPIS.php';
5 require_once 'model/expertWishes.php';
6
7 //Add a wishe of the user if the user click on the button
8 if (is_numeric($_SESSION['id']) && is_numeric($tpiChoosen) && empty($cancelChoice)) {
9     addWish($_SESSION['id'], $tpiChoosen);
10 }
11
12 if (!empty($cancelChoice)) {
13     rmWish($_SESSION['id'], $tpiChoosen);
14 }
15 require_once 'view/tabTPI.php';
```

Listing 5.4 – controller/displayTPI.php

5.5 controller/displayValidationTPI.php

```
1 <?php
2 require_once 'model/crudTPIS.php';
3 require_once 'model/displayTPIValidation.php';
4 require_once 'view/validateTPIs.php';
```

Listing 5.5 – controller/displayValidationTPI.php

5.6 controller/editParam.php

```
1 <?php
2 require_once 'model/formParams.php';
3
4 if ($btn == 'send') {
5     $newDateStart = FILTER_INPUT(INPUT_POST, 'dateStart', FILTER_SANITIZE_STRING);
6     $newDateEnd = FILTER_INPUT(INPUT_POST, 'dateEnd', FILTER_SANITIZE_STRING);
7     $newNbExpMax = FILTER_INPUT(INPUT_POST, 'nbExpMax', FILTER_SANITIZE_STRING);
8
9     updParam(nbExpertMax, $newNbExpMax);
10    updParam(wishesSessionStart, $newDateStart);
11    updParam(wishesSessionEnd, $newDateEnd);
12
13    header('Location: ?action=home');
14    exit;
15 }
16
17 require_once 'view/formParams.php';
```

Listing 5.6 – controller/editParam.php

5.7 controller/home.php

```
1 <?php
2 require_once 'view/home.php' ;
```

Listing 5.7 – controller/home.php

5.8 controller/login.php

```
1 <?php
2 require_once 'model/login.php';
3
4 if ($btn == 'send') {
5     $email = FILTER_INPUT(INPUT_POST, 'inputEmailAddress', FILTER_SANITIZE_STRING);
6     $pwd = FILTER_INPUT(INPUT_POST, 'inputPassword', FILTER_SANITIZE_STRING);
7
8     if (login($email, $pwd)) {
9         header('Location: ?action=home');
10        exit;
11    } else {
12        $error = true;
13    }
14 }
15
16 require_once 'view/login.php';
```

Listing 5.8 – controller/login.php

5.9 controller/logout.php

```
1 <?php
2 require_once 'model/logout.php';
```

Listing 5.9 – controller/logout.php