|  |
| --- |
| Savoir Faires Comportementaux  Radars |
| Magnifique Graphique Radar dans R avec les Packages FMSB et GGPlot -  Datanovia |
| 11 décembre 2023  CEPEGRA  Créé par : Pierre et Grégory |



# INGRWF-11

|  |
| --- |
| Exercice intégré n°1 Dans le cadre de cet exercice, il y a 2 modules à mettre en place :   1. Une API (technologie au choix) 2. Une application cliente pour les stagiaires et les formateurs. |
| *« Le projet se déclinera en 2 axes : formateurs et stagiaires ».*  *Technologies prévues : HTML, CSS, JS (et MySQL - PHP)* |
| Projet L’objectif est de digitaliser cette phase d’évaluation des stagiaires en formation.  La version papier vous est fournie en annexe.  Il s’agit d’évaluer les savoir-faire comportementaux liés à un métier, le métier auquel se forme le stagiaire. Savoir-faire ? Ce sont les comportements liés à des profils professionnels comme l’esprit d’analyse, la motivation, la curiosité, l’autonomie, l’acceptation de la critique, l’organisation,…  Ces savoir faire sont définis pour chaque métier. Un responsable encode une description et 4 situations illustrant un niveau. Cet endage pourrait se faire directement via une interface admin liée à l’API (Dashboard en WP ou Cockpit) et ne fait donc pas partie de cet exercice. Compte stagiaire Le stagiaire doit pouvoir s’autoévaluer jusqu’à 3 fois par formation :   * Il doit donc pouvoir s’inscrire à la plateforme (en version 2, ajouter sa photo). * Il doit pouvoir se connecter à la plateforme via un login et password * Une fois connecté il doit   + voir les formations pour lesquelles il a déjà des évaluations (et voir ses résultats) et la formation en cours et pour laquelle il aura des évaluations.   + pouvoir s’auto-évaluer à une date précisée par le formateur référant de la formation.  Compte formateur Un formateur doit pouvoir évaluer ses stagiaires et voir les auto-évaluations de ses stagiaires.   * Il doit donc pouvoir s’inscrire à la plateforme (en version 2, ajouter sa photo). * Il doit pouvoir se connecter à la plateforme via un login et password. * Une fois connecté, il doit pouvoir :   + Créer des sessions de formations :     - Choisir un métier, centre de formation, date de début et de fin     - Y insérer les stagiaires qui vont su   ivre la formation   * + - Spécifier les dates des gestions évaluations (min. 1 & max. 3)   + Evaluer les stagiaires à chaque gestion évaluation  ÉvaluationPar le stagiaire Lors d’une évaluation, à la demande du formateur référant de la formation (ou sur notification en version 2), le stagiaire se connecte à son compte, retrouve la formation et à accès à l’évaluation liée à la Gestion évaluation.  Une page affiche les comportements liés au métier et demande au stagiaire de se positionner sur chaque comportement (curseur de 1 à 4).  Il valide son évaluation et un graphique type « radar » s’affiche.  Si il ne s’agit pas de la première évaluation de la formation, il peut comparer son évaluation actuelle avec la/les précédente(s) grâce aux Radars.  Si le formateur a donné sa propore évaluation, le stagiaire peut comparer la/les sienne(s) à celle(s) du formateur.  Exemple : <https://codepen.io/pcharli/pen/qBJxgmX> Par le formateur Un fois connecté :   * Un formateur retrouve la liste des formations en cours pour lesquelles il est référant. * Il clique sur une formation :   + Un écran lui indique les dates des GE passées et la date de la prochaine GE.   + Il peut cliquer sur toutes les dates passées pour afficher la liste des stagiaires et leur radar miniature pour cette GE.   + Il peut cliquer sur la date de la prochaine GE et lister ses stagiaires.   + En cliquant sur un stagiaire, il accède au formulaire lui permettant d’évaluer le stagiaire.   + Si un stagiaire a déjà été évalué, un radar miniature s’affiche.  Page « Export » En fin de formation, lors de la dernière GE, le formateur peut exporter les données de toutes les évaluations (toutes les GE, évaluations stagiaires et formateurs) pour les enregistrer en format JSON. Méthodologie  * Git/GitHub * Netlify pour la partie cliente * Hostinger avec déploiement auto pour la partie Backend * Vite.js serait conseillé |