

JOURVIE-SYNTHÈSE

PRÉSENTATION SOMMAIRE

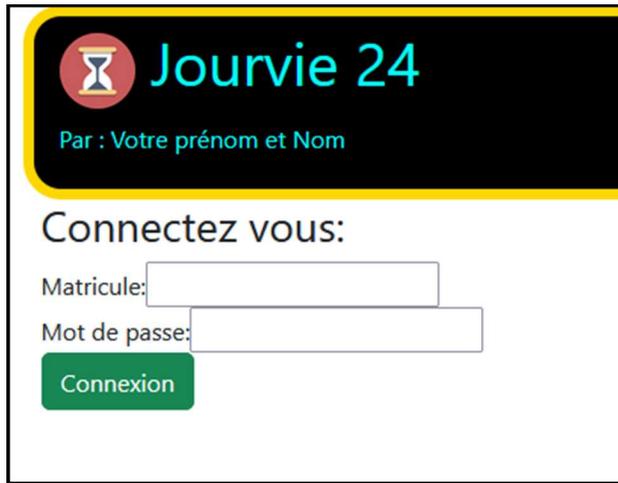
Vous avez le mandat d'ajouter des fonctionnalités à l'application **Jourvie**, application de gestion de projets permettant aux développeurs de comptabiliser facilement le temps consacré aux différentes tâches d'un projet en plus de tenir un journal de leurs avancements.

SPÉCIFICATIONS DU PROGRAMME

Les améliorations et nouveautés que vous apporterez à Jourvie auront deux facettes : client (Angular) et serveur (PHP/MySQL). Le code source de base des deux portions vous est fourni et vous devez le faire évoluer pour y intégrer les fonctionnalités prescrites par le présent énoncé:

Modifier Component App :

1. Afficher votre signature dans la bannière de l'application. Votre signature n'apparaîtra qu'à l'étape pré-connexion et disparaîtra dès qu'un utilisateur est connecté.



- 2.



Date	Heure	num tac	Type
2024-11-01	23:04:24	LOG-5	▶
	23:04:18	LOG-2	⊘
	13:46:44	LOG-2	▶
	13:46:32	LOG-1	⊘

Modifier Component Journal :

1. Ajoutez la récupération et l'affichage des commentaires dans le journal
 - a. Les sources de départ contiennent l'ajout de commentaires et l'affichage de commentaires récents en mémoire vive. Cependant ils ne récupèrent pas les commentaires stockés en BD. Faites en sorte d'afficher tous les commentaires dans le journal :



Journal actif sur M-5

Date	Heure	num tac	Type	Contenu
2024-11-02	11:23:48	M-5	▶	
2024-10-25	23:32:11	Tous	⏻	
	23:32:09		👤	J'ai réglé le problème et on a fini la réunion
	22:51:30		👤	Je dois fix des issues de build+ Réunion
	22:51:14	Tous	▶	
2024-10-19	17:06:29	Tous	⏻	
	17:06:28		👤	On a fait une réunion normale

- b. Les événements du journal, incluant les commentaires, seront affichés par ordre décroissant de date (le plus récent en haut)
 - c. Sur une rangée de commentaire **on n'affiche pas** le numéro de la tâche
2. Dans la zone du statut du journal ajoutez un compteur de temps écoulé depuis le début (ou la fin) de la plus récente session de travail.
 - a. Dès qu'une session de travail commence, vous initialisez l'accumulateur de temps à zéro.
 - b. Il s'incrémente d'une unité par seconde
 - c. Au commencement de la session de travail on utilise les secondes :



Journal actif sur LOG-5, depuis 6.4 secondes

Date	Heure	num tac	Type	Contenu
2024-11-03	06:58:52	LOG-5	▶	
	06:58:29	LOG-2	⏻	

- d. Après 180 secondes, on change l'unité de mesure temporelle pour des minutes :



Journal actif sur LOG-5, depuis 3.1 minutes

Date	Heure	num tac	Type	Contenu
------	-------	---------	------	---------

Après 180 minutes, la granularité passe aux heures.

- e. Le compteur d'activité se rafraichira aux deux secondes. Utilisez `setInterval()` pour contrôler la fréquence de rafraichissement.

Par exemple :

```
ngOnInit(){
  setInterval(
    () => {
      this.augmenteDepuis();
    }, this.delai);
}
```

- f. À chaque nouvelle session de travail, l'accumulateur de temps d'activité se réinitialise à zéro
- g. Le compteur affichera une précision au dixième de son unité de mesure
3. Ajoutez un compteur de temps d'inactivité dans le journal.
- a. Dès qu'un développeur tombe inactif vous initialiserez ce compteur à zéro.



- b. Vous l'afficherez selon les mêmes règles que le compteur d'activité.
- c. Le compteur affichera une précision au dixième de son unité de mesure

Component Admin :

1. Créez un component nommé **admin**
2. Les fonctionnalités de ce component seront accessibles à partir d'une connexion « admin ».
3. Pour se connecter « admin » on fournit le matricule « admin » avec le mot de passe « 11 ». N'insérez pas cet utilisateur en BD, détectez et validez la connexion admin **directement dans le code.**
4. De ce nouveau component on gèrera l'affichage de deux contextes :
 - a. Sessions en cours
 - b. Sommaire développeurs
5. Ces contextes seront accessibles par des boutons. Par défaut on affiche la vue « Sessions en cours » : il s'agit d'afficher toutes les sessions de travail actives de tous les développeurs. Qu'est-ce qu'une session active : **elle n'a pas de date de fin**

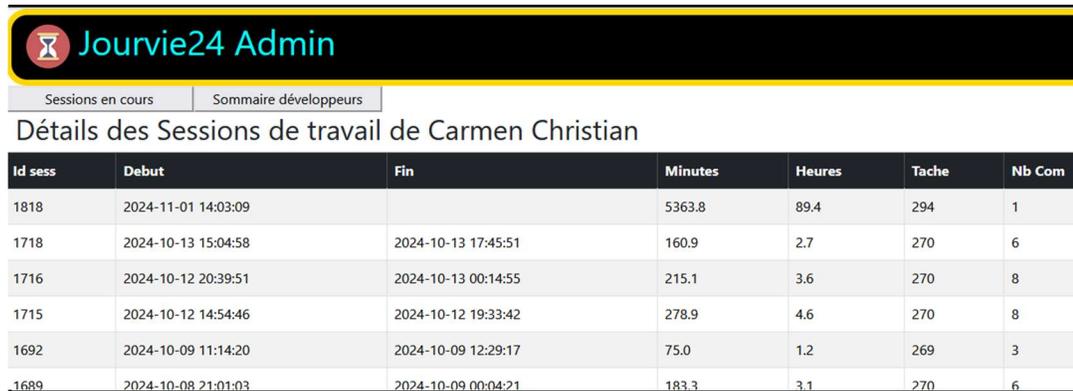
6. Dans le contexte « Sessions en cours » on offre un bouton « Sommaire développeurs » et vice-versa : le bouton représentant le contexte actif n'est pas visible.
7. Pour chacune des sessions de travail en cours, affichez
 - a. L'id du développeur,
 - b. L'id de la session,
 - c. La date de début de la session,
 - d. Le temps écoulé dans cette session en minutes (précision jusqu'au dixième),
 - e. Le temps écoulé dans cette session en heures (précision jusqu'au dixième),
 - f. L'id de la tâche visée par la session
 - g. Le nombre de commentaires dans cette session de travail :

Dev Id	Sess Id	Début	Minutes	Heures	Tache Id	Nb Comm
28	1820	2024-11-04 15:36:05	18.1	0.3	212	1
36	1819	2024-11-04 12:12:12	222.0	3.7	291	0

8. Un clic sur le bouton « Sommaire développeurs » affichera tous les développeurs de la BD. Pour chacun des développeurs on affiche :
 - a. Son matricule, son prénom et son nom
 - b. Le nom du projet assigné et le id de projet
 - c. Le nombre de sessions commencées par le développeur
 - d. Le nombre total des heures réalisées dans ses sessions (précision au dixième)
 - e. Le nombre de commentaires rédigés par le développeur
 - f. Le tableau est trié en ordre décroissant du nombre d'heures réalisées

Mat + Nom	Projet (id)	Nb Sess Trav	Heures	Nb Comm
1111111 + Antoine Allard	Exquizz (8)	35	857.3	98
8888888 + Homère Homerun	GamAmer (9)	37	467.1	93
11 + Louise Léonard	Niuzz (10)	59	443.1	147
0000000 + Keros Kanga	Niuzz (10)	41	419.9	273
2222222 + Bugeault Béatrice	Exquizz (8)	39	291.3	100
7777777 + Gratien Gélinas	GamAmer (9)	42	283.4	135
5555555 + Emilo Estérela	GamAmer (9)	48	254.6	188
3333333 + Christian Carmen	Exquizz (8)	25	139.2	126

9. Un clic sur le matricule du développeur ouvrira le contexte « Détails d'un développeur »
 - a. Sur cet écran on affichera toutes les sessions réalisées par le développeur sélectionné. Pour chaque session du développeur on affichera
 - i. L'Id de la session de travail
 - ii. La date de début
 - iii. La date de fin
 - iv. Le nombre de minutes réalisées (précision au dixième)
 - v. Le nombre d'heures réalisées (précision au dixième)
 - vi. L'Id de la tâche visée par la session de travail
 - vii. Le nombre de commentaires rédigés pendant cette session
 - viii. Le tableau est trié en ordre décroissant de Début : la plus récente en haut



The screenshot shows the 'Jourvie24 Admin' interface. At the top, there are two buttons: 'Sessions en cours' and 'Sommaire développeurs'. Below them is the title 'Détails des Sessions de travail de Carmen Christian'. A table lists the sessions with columns for 'Id sess', 'Debut', 'Fin', 'Minutes', 'Heures', 'Tache', and 'Nb Com'.

Id sess	Debut	Fin	Minutes	Heures	Tache	Nb Com
1818	2024-11-01 14:03:09		5363.8	89.4	294	1
1718	2024-10-13 15:04:58	2024-10-13 17:45:51	160.9	2.7	270	6
1716	2024-10-12 20:39:51	2024-10-13 00:14:55	215.1	3.6	270	8
1715	2024-10-12 14:54:46	2024-10-12 19:33:42	278.9	4.6	270	8
1692	2024-10-09 11:14:20	2024-10-09 12:29:17	75.0	1.2	269	3
1689	2024-10-08 21:01:03	2024-10-09 00:04:21	183.3	3.1	270	6

- ix. Les deux boutons « Sessions en cours » et « Sommaire développeurs » sont alors visibles

LE CODE DE DÉPART

Vous pouvez démarrer votre TP à l'aide de deux projets sur github.com aux URL suivantes :

<https://github.com/alainprof/jv24> //portion client Angular

<https://github.com/alainprof/jv24-srv> // portion serveur PHP et script SQL de la BD

LA BASE DE DONNÉES

Le fichier **jv24-synthese.sql**, disponible dans le projet github de la portion serveur, est un script SQL permettant d'installer la BD nécessaire au TP. Vous pouvez utiliser une BD en localhost pour la phase de développement, mais utilisez votre espace sur le serveur **techinfo-cstj.ca** pour la mise en production.

Il y a cinq tables : **projets**, **taches**, **developpeurs**, **sessionstravail** et **commentaires**, peuplées avec de vraies données. Leurs structures ressemblent à ce qu'on a vu pendant le cours et vous pouvez les consulter dans le script d'installation.

EXIGENCES GÉNÉRALES

1. Vos classes, méthodes et attributs doivent porter des noms significatifs
2. Votre code doit être bien commenté (en-tête de fichiers, séparateurs de méthodes et code complexe)
3. Les données doivent être stockées dans une BD installée sur votre sous-domaine du serveur techinfo-cstj.ca
4. Les données doivent être récupérées par un service Angular
5. Les URL utilisées par le service doivent correspondre à des scripts PHP
6. Ces scripts PHP récupéreront les données grâce à l'API mysqli et les transmettront au client sous forme de JSON

EXIGENCES DE REMISE

1. Il y a cinq niveaux de remises de votre projet :
 - i. Version en **production** sur le serveur du département (techinfo-cstj.ca) accessible à partir de la rubrique « Sites des étudiants » de ma page personnelle (<https://amartel.techinfo-cstj.ca/4203D4/>)
 - ii. Remise de la portion Angular sur **LÉA : ne remettez que le dossier src**;
 - iii. Remise de la portion PHP sur LÉA : remettez tous les fichiers sources PHP, même ceux que vous n'avez pas modifiés
 - iv. **Projet Angular sur Github.com** : hébergez votre projet dans un repo privé sur github.com et remettez sur LÉA un fichier texte donnant **l'URL de votre repo-client**. Ajoutez-moi (AlainProf) comme collaborateur de ce repo.
 - v. **Fichiers PHP sur Github.com** : hébergez la portion PHP de votre projet dans un repo privé sur github.com et remettez sur LÉA un fichier texte donnant **l'URL de votre repo-serveur**. Ajoutez-moi (AlainProf) comme collaborateur de ce repo.
2. Ce travail doit être réalisé individuellement. Pas de travail d'équipe.

ÉVALUATION

Ce travail pratique compte pour 30% de votre note finale.

Pondération des éléments :

Élément	Valeur
Votre signature	2
Affichage des commentaires	13
Compteur de temps actif	20
Compteur de temps inactif	10
Connexion Admin	5
Admin "sessions en cours"	20
Admin "sommaire dev"	15
Admin "Détails dev"	15
Respect des exigences	-40
Total	100

La date limite de remise est le mardi 11 décembre 2024 avant 23h59. Pénalités de 10% par jour de retard.