```
<?php
/*
 * Auteur       : Hoarau Nicolas
 * Projet       : WE GO
 * Page         : getEvents.php
 * Date         : 28.05.2020
 * Description  : Fichier qui récupère les événements pour la gestion des événements s
on est connecté sinon récupère les événements publiques.
 * Version      : 1.0 : Récupère les événements pour la gestion des événements si on
est connecté
 */

require_once dirname(__DIR__) . '/Controller/DatabaseController.php';
require_once __DIR__ . './backend.php';

if (session_status() == PHP_SESSION_NONE) {
  session_start();
}

$dateMoreOneHour = date('Y-m-d H:i:s', strtotime(date('Y-m-d H:i:s') . '+1hours'));
$idUser = $_SESSION['loggedIn']['idUser'];

$queryGetEventsParticpating = <<<EX
SELECT e.idEvenement, e.nom, e.descriptif, e.dateDebut, e.dateFin, e.urlImage, u.pseuc
FROM inscriptions AS i
JOIN evenement AS e ON i.idEvenement = e.idEvenement
JOIN user AS u ON e.idOrganisateur = u.idUser
WHERE i.idUser = :idUser
AND e.idOrganisateur <> :idUser
AND TIMESTAMP(dateDebut) >= :dateTime
EX;

$queryGetPublicEventsNotParticipating = <<<EX
SELECT e.idEvenement, e.nom, e.descriptif, e.dateDebut, e.dateFin, e.urlImage, u.pseuc
FROM evenement AS e
JOIN user AS u ON u.idUser = e.idOrganisateur
WHERE idEvenement NOT IN (
  SELECT idEvenement
  FROM inscriptions
  WHERE idUser = :idUser )
AND prive = 0
AND e.idOrganisateur <> :idUser;
EX;

$queryGetInvitedEvents = <<<EX
SELECT e.idEvenement, e.nom, e.descriptif, e.dateDebut, e.dateFin, e.urlImage, u.pseuc
FROM invites AS i
JOIN evenement AS e ON e.idEvenement = i.idEvenement
JOIN user AS u ON u.idUser = e.idOrganisateur
WHERE prive = 1
AND TIMESTAMP(dateDebut) >= :dateTime
AND e.idEvenement NOT IN (
  SELECT idEvenement
  FROM inscriptions
```

```php
  WHERE idUser = :idUser )
AND i.idUser = :idUser
ORDER BY dateDebut DESC;
EX;

$queryGetEventCreated = <<<EX
SELECT e.idEvenement, e.nom, e.descriptif, e.dateDebut, e.dateFin, e.urlImage, u.pseuc
FROM evenement AS e
JOIN user AS u ON u.idUser = e.idOrganisateur
WHERE e.idOrganisateur = :idUser
AND TIMESTAMP(dateDebut) >= :dateTime
EX;

try {
  $requestGetPublicEventsNotParticipating = DatabaseController::prepare($queryGetPublicEventsNotParticipating);
  $requestGetPublicEventsNotParticipating->bindParam(':idUser', $idUser, PDO::PARAM_INT, 11);
  $requestGetPublicEventsNotParticipating->bindParam(':dateTime', $dateMoreOneHour);

  $requestGetPublicEventsNotParticipating->execute();
  $result['eventsNotParticipating'] = $requestGetPublicEventsNotParticipating->fetchAll(PDO::FETCH_ASSOC);

  $requestGetInvitedEvents = DatabaseController::prepare($queryGetInvitedEvents);
  $requestGetInvitedEvents->bindParam(':idUser', $idUser, PDO::PARAM_INT, 11);
  $requestGetInvitedEvents->bindParam(':dateTime', $dateMoreOneHour);

  $requestGetInvitedEvents->execute();
  $result['invitedEvents'] = $requestGetInvitedEvents->fetchAll(PDO::FETCH_ASSOC);

  $requestGetEventsParticpating = DatabaseController::prepare($queryGetEventsParticpating);
  $requestGetEventsParticpating->bindParam(':idUser', $idUser, PDO::PARAM_INT, 11);
  $requestGetEventsParticpating->bindParam(':dateTime', $dateMoreOneHour);

  $requestGetEventsParticpating->execute();
  $result['eventsParticipating'] = $requestGetEventsParticpating->fetchAll(PDO::FETCH_ASSOC);

  $requestGetEventCreated = DatabaseController::prepare($queryGetEventCreated);
  $requestGetEventCreated->bindParam(':idUser', $idUser, PDO::PARAM_INT, 11);
  $requestGetEventCreated->bindParam(':dateTime', $dateMoreOneHour);

  $requestGetEventCreated->execute();
  $result['eventsCreated'] = $requestGetEventCreated->fetchAll(PDO::FETCH_ASSOC);

  $result['isLogged'] = true;
  $result['nickname'] = $_SESSION['loggedIn']['pseudo'];

  echo json_encode($result);
  exit();
} catch (PDOException $e) {
  echo json_encode([
    'ReturnCode' => 1,
    'Error' => 'Erreur est survenue lors de la récupération des événements'
  ]);
  exit();
}

```