



HAUTE ECOLE
Provinciale de
H A I N A U T

CONDORCET

Projet de fin d'année

RAPPORT TECHNIQUE

3ème Bachelier INFORMATIQUE DE GESTION

COURS DE : PROGRAMMATION AVANCEE

MAÎTRE-ASSISTANT : S. DELUCA

Etudiant : Marghem Jeremy

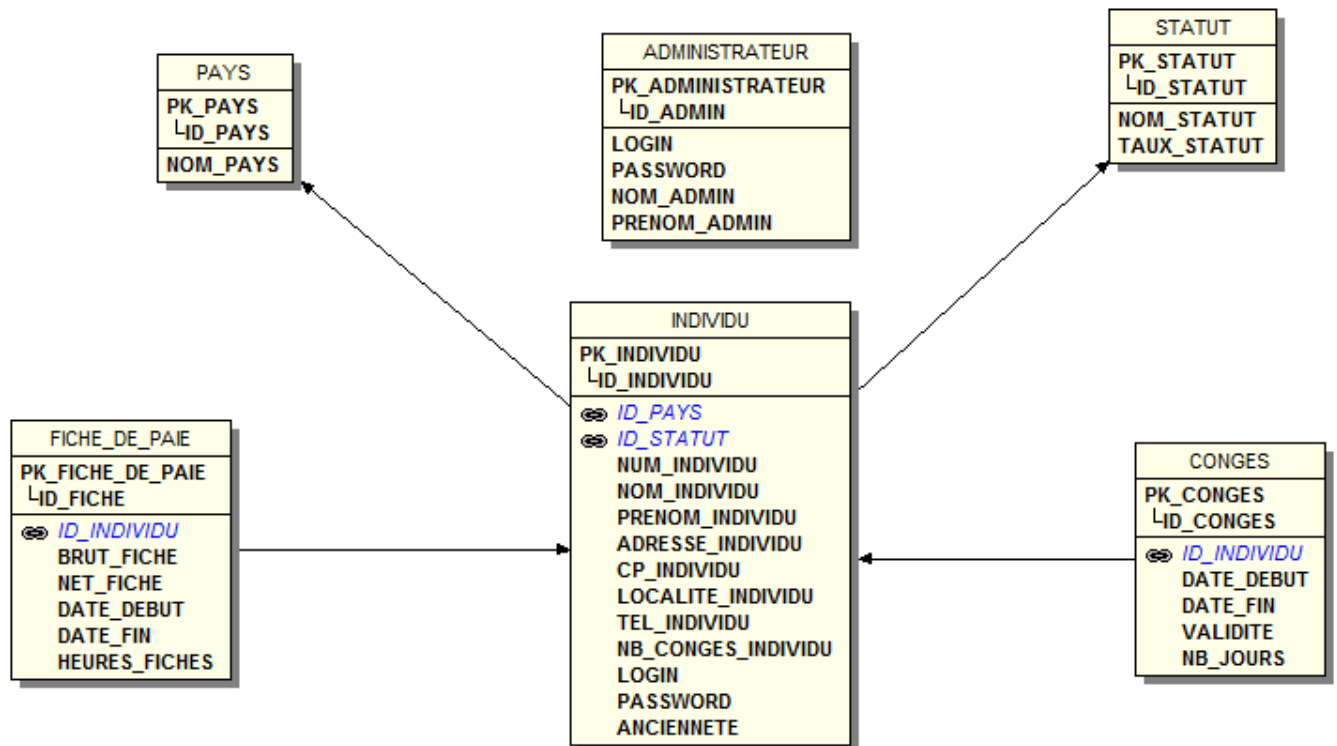


Enseignement
organisé par la



Base de données

MCD



- Un INDIVIDU appartient à un et un seul pays
- Un INDIVIDU possède à un et un seul statut
- Une FICHE DE PAIE appartient à un et un seul INDIVIDU
- Un CONGES appartient à un et un seul INDIVIDU
- La table ADMINISTRATEUR est indépendante

SCRIPT DE CREATION

```
-- -----
--      Génération d'une base de données pour
--      Oracle Version 9i
--      (6/1/2017 11:55:00)
-- -----
--      Auteur : Jeremy
--      Date de dernière modification : 6/1/2017 11:49:03
-- -----

DROP TABLE CONGES CASCADE CONSTRAINTS;

DROP TABLE STATUT CASCADE CONSTRAINTS;

DROP TABLE INDIVIDU CASCADE CONSTRAINTS;

DROP TABLE ADMINISTRATEUR CASCADE CONSTRAINTS;

DROP TABLE PAYS CASCADE CONSTRAINTS;

DROP TABLE FICHE_DE_PAIE CASCADE CONSTRAINTS;

-- -----
--      TABLE : CONGES
-- -----

CREATE TABLE CONGES
(
  ID_CONGES NUMBER(2) NOT NULL,
  ID_INDIVIDU NUMBER(2) NOT NULL,
  DATE_DEBUT VARCHAR2(128) NOT NULL,
  DATE_FIN VARCHAR2(128) NOT NULL,
  VALIDITE NUMBER(2) NOT NULL,
  NB_JOURS NUMBER(2) NOT NULL
, CONSTRAINT PK_CONGES PRIMARY KEY (ID_CONGES)
);

-- -----
--      INDEX DE LA TABLE CONGES
-- -----

CREATE INDEX I_FK_CONGES_INDIVIDU
  ON CONGES (ID_INDIVIDU ASC)
;
```

-- TABLE : STATUT

```
CREATE TABLE STATUT
(
  ID_STATUT NUMBER(2) NOT NULL,
  NOM_STATUT CLOB NOT NULL,
  TAUX_STATUT NUMBER NOT NULL
, CONSTRAINT PK_STATUT PRIMARY KEY (ID_STATUT)
);
```

-- TABLE : INDIVIDU

```
CREATE TABLE INDIVIDU
(
  ID_INDIVIDU NUMBER(2) NOT NULL,
  ID_PAYS NUMBER(2) NOT NULL,
  ID_STATUT NUMBER(2) NOT NULL,
  NUM_INDIVIDU CLOB NOT NULL,
  NOM_INDIVIDU CLOB NOT NULL,
  PRENOM_INDIVIDU CLOB NOT NULL,
  ADRESSE_INDIVIDU CLOB NOT NULL,
  CP_INDIVIDU CLOB NOT NULL,
  LOCALITE_INDIVIDU CLOB NOT NULL,
  TEL_INDIVIDU CLOB NOT NULL,
  NB_CONGES_INDIVIDU NUMBER(2) NOT NULL,
  LOGIN CLOB NOT NULL,
  PASSWORD CLOB NOT NULL,
  ANCIENNETE NUMBER(2) NOT NULL
, CONSTRAINT PK_INDIVIDU PRIMARY KEY (ID_INDIVIDU)
);
```

-- INDEX DE LA TABLE INDIVIDU

```
CREATE INDEX I_FK_INDIVIDU_PAYS
ON INDIVIDU (ID_PAYS ASC)
;
```

```
CREATE INDEX I_FK_INDIVIDU_STATUT
ON INDIVIDU (ID_STATUT ASC)
;
```

-- TABLE : ADMINISTRATEUR

```
CREATE TABLE ADMINISTRATEUR
(
  ID_ADMIN NUMBER(2) NOT NULL,
  LOGIN NUMBER(2) NOT NULL,
  PASSWORD NUMBER(2) NOT NULL,
  NOM_ADMIN CLOB NOT NULL,
  PRENOM_ADMIN CLOB NOT NULL
, CONSTRAINT PK_ADMINISTRATEUR PRIMARY KEY (ID_ADMIN)
);
```

-- TABLE : PAYS

```
CREATE TABLE PAYS
(
  ID_PAYS NUMBER(2) NOT NULL,
  NOM_PAYS CLOB NOT NULL
, CONSTRAINT PK_PAYS PRIMARY KEY (ID_PAYS)
);
```

-- TABLE : FICHE_DE_PAIE

```
CREATE TABLE FICHE_DE_PAIE
(
  ID_FICHE NUMBER(2) NOT NULL,
  ID_INDIVIDU NUMBER(2) NOT NULL,
  BRUT_FICHE NUMBER NOT NULL,
  NET_FICHE NUMBER NOT NULL,
  DATE_DEBUT DATE NOT NULL,
  DATE_FIN DATE NOT NULL,
  HEURES_FICHES NUMBER(2) NOT NULL
, CONSTRAINT PK_FICHE_DE_PAIE PRIMARY KEY (ID_FICHE)
);
```

-- INDEX DE LA TABLE FICHE_DE_PAIE

```
CREATE INDEX I_FK_FICHE_DE_PAIE_INDIVIDU
  ON FICHE_DE_PAIE (ID_INDIVIDU ASC)
;
```

-- CREATION DES REFERENCES DE TABLE

```
ALTER TABLE CONGES ADD (
  CONSTRAINT FK_CONGES_INDIVIDU
  FOREIGN KEY (ID_INDIVIDU)
  REFERENCES INDIVIDU (ID_INDIVIDU)) ;
```

```
ALTER TABLE INDIVIDU ADD (
  CONSTRAINT FK_INDIVIDU_PAYS
  FOREIGN KEY (ID_PAYS)
  REFERENCES PAYS (ID_PAYS)) ;
```

```
ALTER TABLE INDIVIDU ADD (
  CONSTRAINT FK_INDIVIDU_STATUT
  FOREIGN KEY (ID_STATUT)
  REFERENCES STATUT (ID_STATUT)) ;
```

```
ALTER TABLE FICHE_DE_PAIE ADD (
  CONSTRAINT FK_FICHE_DE_PAIE_INDIVIDU
  FOREIGN KEY (ID_INDIVIDU)
  REFERENCES INDIVIDU (ID_INDIVIDU)) ;
```

-- FIN DE GENERATION

FONCTIONS POSTGRESQL

Fonction permettant de créer un nouvel individu

```
-- Function: createindividu(integer, integer, text, text, text, text, text, text, text, integer, text, text, integer)
```

```
-- DROP FUNCTION createindividu(integer, integer, text, text, text, text, text, text, text, integer, text, text, integer);
```

```
CREATE OR REPLACE FUNCTION createindividu(  
    integer,  
    integer,  
    text,  
    text,  
    text,  
    text,  
    text,  
    text,  
    text,  
    integer,  
    text,  
    text,  
    integer)  
    RETURNS integer AS  
$BODY$  
declare  
    retour integer;  
begin  
    INSERT INTO INDIVIDU  
(id_pays,id_statut,num_individu,nom_individu,prenom_individu,adresse_individu,cp_individu,localite_individu,tel_individu,nb_conges_individu,login,password,anciennete) VALUES  
($1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13);  
    retour=1;  
    return retour;  
end;  
$BODY$  
LANGUAGE plpgsql VOLATILE  
COST 100;  
ALTER FUNCTION createindividu(integer, integer, text, text, text, text, text, text, text, integer, text, text, integer)  
OWNER TO "jeremy.marghem";
```

Fonction permettant de créer une nouvelle demande de congés

```
-- Function: createconge(date, date, integer, integer, integer)

-- DROP FUNCTION createconge(date, date, integer, integer, integer);

CREATE OR REPLACE FUNCTION createconge(
    date,
    date,
    integer,
    integer,
    integer)
    RETURNS integer AS
$BODY$
declare
    retour integer;
begin
    INSERT INTO conges (date_debut,date_fin,nb_jours,id_individu,validite) VALUES ($1,$2,$3,$4,$5);
    retour=1;
    return retour;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION createconge(date, date, integer, integer, integer)
    OWNER TO "jeremy.marghem";
```


Fonction permettant de créer une nouvelle fiche de paie

```
-- Function: createfiche(date, integer, date, real, real, integer)

-- DROP FUNCTION createfiche(date, integer, date, real, real, integer);

CREATE OR REPLACE FUNCTION createfiche(
    date,
    integer,
    date,
    real,
    real,
    integer)
    RETURNS integer AS
$BODY$
declare
    retour integer;
begin
    INSERT INTO FICHE_DE_PAIE (date_debut,id_individu,date_fin,brut_fiche,net_fiche,heures_fiche)
VALUES ($1,$2,$3,$4,$5,$6);
    retour=1;
    return retour;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION createfiche(date, integer, date, real, real, integer)
    OWNER TO "jeremy.marghem";
```

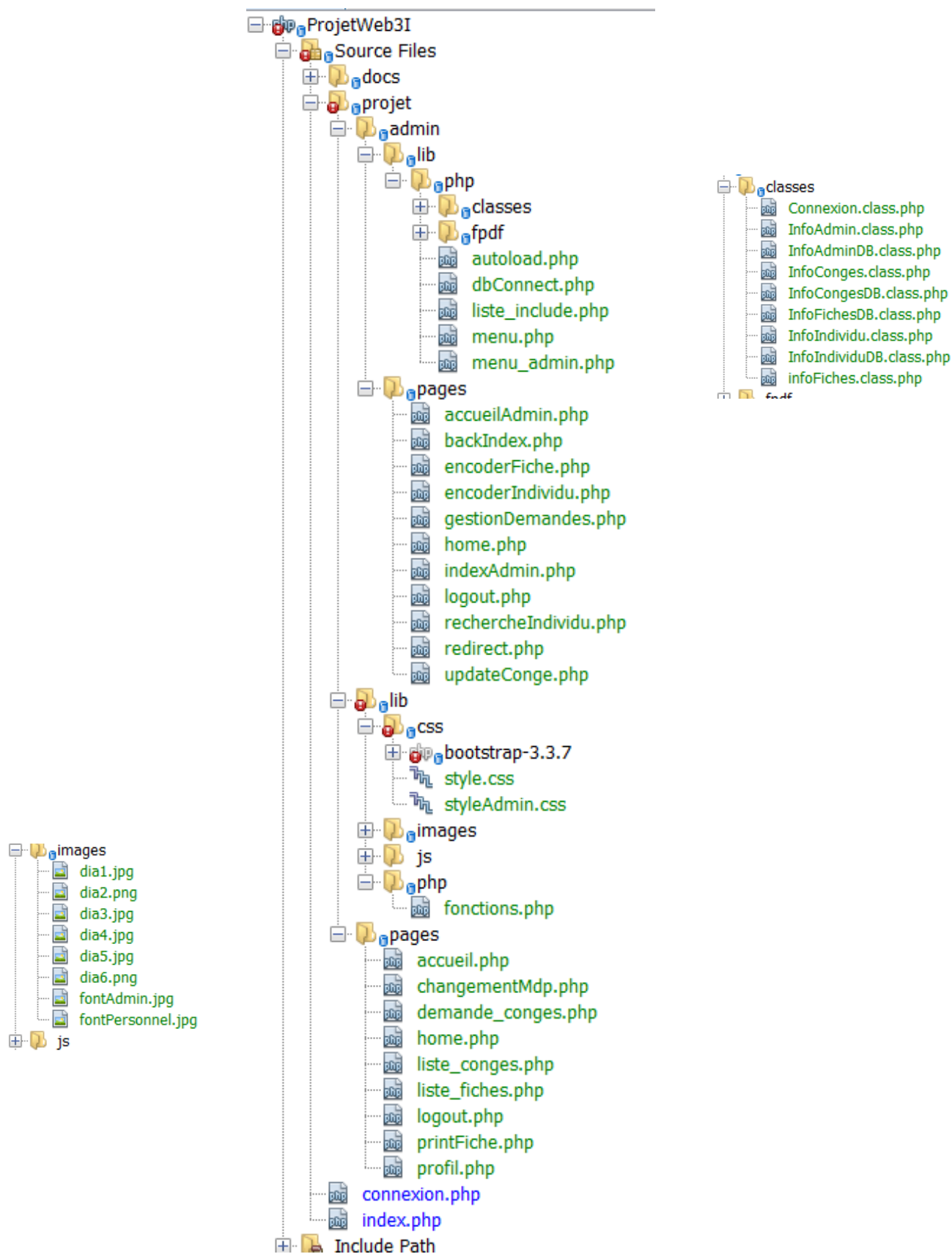
Fonction permettant de mettre à jour une demande de congés

```
-- Function: updateconge(integer, integer)

-- DROP FUNCTION updateconge(integer, integer);

CREATE OR REPLACE FUNCTION updateconge(
    integer,
    integer)
    RETURNS integer AS
$BODY$
DECLARE
    declare f_id alias for $1;
    declare f_val alias for $2;
    declare retour integer;
    declare id integer;
BEGIN
    SELECT INTO id id_conges FROM conges WHERE id_conges=f_id;
    IF NOT FOUND
    THEN
        retour = 0;
    ELSE
        UPDATE conges SET valide=f_val WHERE id_conges=f_id;
        retour = 1;
    END IF;
    return retour;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION updateconge(integer, integer)
    OWNER TO "jeremy.marghem";
```

ARBORESCENCE DU PROJET



DESCRIPTION DES FICHIERS

LES CLASSES

Connexion.class.php

Classe de Connexion permettant d'avoir une seule instance de connexion à la fois

InfoAdmin.class.php

Classe Admin utilisée par InfoAdminDB.class.php

InfoAdminDB.class.php

Classe permettant d'accéder à la base de données.

Methodes :

-getVerifAdmin(\$login,\$password)

-> Permet de vérifier les identifiants lors d'une tentative de connexion sur la page administration

InfoConges.class.php

Classe Conges utilisée par InfoCongesDB.class.php

InfoCongesDB.class.php

Classe permettant d'accéder à la base de données

Methodes :

-getInfoConges(\$id)

-> Permet de récupérer une liste de congés appartenant à un individu

-getDemandes()

-> Permet de récupérer une liste de congés en attente de validation

-updateConge(\$id, \$val)

-> Permet de mettre à jour la validité d'un congé

InfoFiches.class.php

Classe Fiches utilisé par InfoFichesDB.class.php

InfoFichesDB.class.php

Classe permettant d'accéder à la base de données

Methodes :

-getInfoFiches(\$id)

-> Permet de récupérer une liste de fiches appartenant à un individu

-getInfoFichesParIdFiche(\$id_fiche)

-> Permet de récupérer une fiche particulière pour générer un PDF

-create(\$debut,\$id,\$fin,\$brut,\$net,\$heures)

-> Permet de créer une nouvelle fiche

InfoIndividu.class.php

Classe Individu utilisée par InfoIndividu.class.php

InfoIndividuDB.class.php

Classe permettant d'accéder à la base de données

Methodes :

-getAllIndividu()

-> Permet de récupérer une liste des individus triée par nom

-getVerifIndividu(\$login, \$password)

-> Permet de vérifier les identifiants lors d'une tentative de connexion sur la page personnel

- getVerifMdp(\$id, \$mdp)

-> Permet de vérifier si le mot de passe entré est correct lors de la maj du mot de passe

- getMajMdp(\$id, \$password, \$newpassword)

-> Permet de mettre à jour un mot de passe appartenant à un individu

- create(\$id_pays,\$id_statut,\$matricule,\$nom,\$prenom,\$adresse,\$cp,\$localite,\$tel,\$conges,\$mdp)

-> Permet de créer un nouvel individu

- augmenteNb(\$id,\$n)

-> Permet d'incrémenter le nombre de congés d'un individu si sa demande de congé est refusé

- diminuerNb(\$id,\$n)

-> Permet de diminuer le nombre de congés d'un individu lors de l'encodage d'une demande

Les pages

Connexion.php

Fichier permettant de se connecter à la base de données

Index.php

Page d'accueil réservée au personnel et contenant un formulaire de connexion.

Dans la partie php, on vérifie si l'utilisateur entré est correct et on instancie des variables de SESSION.

Un script permet quand à lui de vérifier si la série de touches entrées correspond à « admin » qui permet de se diriger vers la page d'accueil réservée aux administrateurs.

Les pages concernant le personnel

Accueil.php

Page principale dans laquelle on viendra insérer les pages requises.

Au lancement, la page d'accueil contiendra la page HOME.

Home.php

Page contenant un carrousel dans laquelle on insère différentes images en dur.

Profil.php

Page affichant les différentes variables de SESSION de l'utilisateur et permettant d'afficher une fiche de profil.

ChangementMdp.php

Formulaire permettant de changer le mot de passe.

On vérifie si le mot de passe actuel est correct et on vérifie si les deux nouveaux mots de passe sont identiques.

Si c'est le cas, on modifie le mot de passe de l'utilisateur dans la base de données.

Liste_conges.php

Page affichant un tableau récapitulatif des demandes de congés de l'utilisateur. En fonction de la validité, la ligne prendra la couleur rouge, orange ou verte.

Demande_conges.php

Page contenant un formulaire d'encodage de demande. Celui-ci contient deux champs dates permettant d'encoder les dates de debut et de fin.

Ces champs seront reliées à un script qui permettant de faire apparaitre un calendrier datepicker pour la selection. Lors de l'encodage, le nombre de jours demandés sera deduit du total de nombre de jours de congés possédés par l'individu.

Liste_fiches.php

Page contenant un tableau recapitulatif des fiches de paies. Chaque ligne contient un renvoyant vers printFiche.php qui permet de générer un PDF

Printfiches.php

Page générant un PDF via FPDF. On y recupere les variables de sessions de l'utilisateur et à l'aide de l'id de la fiche on recupere ses informations.

Logout.php

Petit fichier php permettant de detruire la session en cours et de rediriger vers la page de connexion.

Les pages concernant les administrateurs

AccueilAdmin.php

Page principal dans laquelle on viendra insérer les pages requises.

Home.php

Page affichant un carrousel dont les images sont encodées en dur.

EncoderIndividu.php

Page contenant un formulaire permettant d'encore un nouvel individu.

Les champs code postal, téléphone et nombre de congés sont verifiés via des regex.

RechercheIndividu.php

Page affichant un tableau recapitulatif de toutes les personnes de la base de données.

Deux input permettront de filtrer celui-ci via le matricule et via le nom.

Contient une fonction searchInTable(numero de colonne) permettant de filtrer le contenu des qu'on entre une valeur dans un des deux champs.

EncoderFiche.php

Page affichant un formulaire permettant d'encoder une nouvelle fiche
Le select utilisé est un selectpicker provient du plugin bootstrap select.

GestionDemandes.php

Page affichant un recapitulatif de toutes les demandes de congés ainsi que le nom du demandeur.
Deux boutons permettant de valider ou de refuser une demande. Ceux-ci renvoient vers la page updateConge qui s'occupe du traitement.

UpdateConge.php

Traitement de la validation ou non. Si la demande est refusée, le nombre de congés de l'individu sera augmenter du nombre de jours compris dans la demande.
Un bouton « Ok » permettra quand à lui de revenir directement sur la page de gestion des demandes en passant par le fichier redirect.php.

Logout.php

Page php détruisant la session en cours et redirigeant vers la page de connexion des administrateurs.

BackIndex.php

Fichier php utilisé lors du clic sur le bouton retour situer sur la page de connexion et qui renvoie vers la page de connexion du personnel