Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Arti Mestieri Trevano |
| Data | 29.11.2019 |

|  |
| --- |
| Lavori svolti |
| Oggi ho continuato l’implementazione degli abitanti, ho modifato alcune funzioni e paramatri del habitantModel, questo è il metodo che uso per riempire la tabella iniziale:  public function getAllHabitantForTank($tankName) {  $selectHabitants = "select *\** from abitante where nome\_vasca=:tankName";  $this->statement = $this->connection->prepare($selectHabitants);  $this->statement->bindParam(':tankName', $tankName, PDO::*PARAM\_STR*);  $this->statement->execute();  $result = $this->statement->fetchAll(PDO::*FETCH\_ASSOC*);  return $result; }  Questa funzione la uso per ricavare se esiste già un abitante della stessa specie e dello stesso sesso per la convalidazione e la modifica degli abitanti.  public function getAllHabitantBySpeciesAndSex($species, $sex) {  $selectHabitants = "Select *\** FROM abitante WHERE specie=:species AND genere=:sex";  $this->statement = $this->connection->prepare($selectHabitants);  $this->statement->bindParam(':species', $species, PDO::*PARAM\_STR*);  $this->statement->bindParam(':sex', $sex, PDO::*PARAM\_STR*);  $this->statement->execute();  $result = $this->statement->fetchAll(PDO::*FETCH\_ASSOC*);  return $result; }  Ho implementato il metodo per rimuovere un abitante:  public function delete($species, $sex) {  $selectHabitants = "DELETE FROM abitante WHERE specie=:species AND genere=:sex";  $this->statement = $this->connection->prepare($selectHabitants);  $this->statement->bindParam(':species', $species, PDO::*PARAM\_STR*);  $this->statement->bindParam(':sex', $sex, PDO::*PARAM\_STR*);  $this->statement->execute(); }  Ho modificato anche l’index e ho aggiunto una nuova funzione che uso di default:  Infatti se per caso l’URL punta all’index il sito ti riporta al login perché non ci puoi arrivare dal sito.  public function index(){  *header*("Location:".*URL*); }  Questo invece è il metodo che mostra la pagina della gestione degli abitanti per ogni vasca:  function showAllHabitantsTank($referenceTankName = 1){  $path = *URL* . "habitantManagement/addHabitant";  if($\_SESSION["authentification"] == true && *strlen*($referenceTankName) > 0 && $referenceTankName != 1){  if($this->tankValidation->validateTankName($referenceTankName)) {  $\_SESSION["referencesTankName"] = $referenceTankName;  $habitants = $this->habitantModel->getAllHabitantForTank($\_SESSION["referencesTankName"]);  $nameButton = "Aggiungi";  require "GestioneAcquariMarini/views/\_templates/header.php";  require "GestioneAcquariMarini/views/\_templates/menu.php";  require "GestioneAcquariMarini/views/gestioneAcquari/habitants/index.php";  require "GestioneAcquariMarini/views/\_templates/footer.php";  }else{  *header*("Location:".*URL*);  }  }else{  *header*("Location:".*URL*);  } }  Poi ho fatto la convalidazione nell’aggiunta e l’implementazione, queste sono le funzioni:  public function validateSex($sex){  if ($sex == self::*MALE\_SEX* || $sex == self::*FEMALE\_SEX* || $sex == self::*OTHER\_SEX*) {  return true;  } else {  return false;  } }  public function validatePrimaryKeysHabitant($species, $sex){  $this->allHabitants = $this->habitantModel->getAllHabitantBySpeciesAndSex();  if(*count*($this->allHabitants) > 0){  return true;  }  return false; }  Poi come per gli altri ho implementato la convalidazione dove uso queste funzioni per la convalidazione della specie uso la convalidazione del nome della vasca che è la stessa cosa. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| - |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In ritardo di 4 ore. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| -Finire la gestione degli abitanti delle vasche |