

- **PHP scripts provided for CRUD operations.**

**C-Create:** The query below allows the user to create a new activity in the database. This is from addActivity.php:

```
$query = 'INSERT INTO activities
          (categoryID, babyID, userID, time, description)
          VALUES
          (:categoryID, :babyID, :userID, :time, :description)';
$stmt = $db->prepare($query);
$stmt->bindValue(':categoryID', $categoryID);
$stmt->bindValue(':babyID', $babyID);
$stmt->bindValue(':userID', $userID);
$stmt->bindValue(':time', $time);
$stmt->bindValue(':description', $description);
$stmt->execute();
```

This is the code for creating a user account. This is from register.php:

```
INSERT INTO users (username, passwordHash, fullName)
VALUES (:username, :passwordHash, :fullName);
```

This is the code for creating a doctor account. This is from registerDoctorForm.php:

```
INSERT INTO doctor (username, passwordHash, fullName, type)
VALUES (:username, :passwordHash, :fullName, :type);
```

**R-Read:** This query lets the user read or view all the activities in the database from that specific user. This is from index.php:

```
$queryProducts = 'SELECT a.*, b.babyName
                  FROM activities a
                  JOIN babies b ON a.babyID = b.babyID
                  WHERE a.userID = :user_id
                  AND a.babyID = :baby_id
                  AND a.categoryID = :category_id
                  ORDER BY a.activityID';
```

This code/query lets the user see the baby. This is from index.php:

```
$query = 'SELECT babyID
          FROM babies
          WHERE userID = :user_id
          LIMIT 1';
```

This code/query lets the user see the categories. This is from index.php:

```
$query = 'SELECT *
          FROM categories
          ORDER BY categoryID';
```

**U-Update:** This query lets the user update of the previous activities, if they need to make any edits or changes. This code is from updateActivities.php:

```
$query = 'UPDATE activities  
        SET categoryID = :categoryID, babyID = :babyID,  
            userID = :userID, time = :time, description = :description  
        WHERE activityID = :activityID';
```

**D-Delete:** This query lets the user delete any of the activities from the database at any given time ,it is from deleteActivity.php:

```
$query = 'DELETE FROM activities  
        WHERE activityID = :activityID';  
$statement = $db->prepare($query);  
$statement->bindValue(':activityID', $activityID);  
$statement->execute();
```

#### • Each script is clearly linked to specific user stories.

User Stories for Kazeems Kloud:

#### User Stories for [User - Parents/Guardians]

1. Story 1: As a parent/guardian, I should be able to create an account so that I can store my child's information on the website for reference.

Associated Scripts: login.php and register.php

2. Story 2: As a parent/guardian, I should be able to log a child's feeding details (type, quantity, time) so that I can monitor what they eat.

Associated Scripts: index.php, addActivity.php and addActivityForm.php

3. Story 3: As a parent/guardian, I should be able to log a child's drinking details (type, quantity, time) so that I can monitor what they drink.

Associated Scripts: index.php, addActivity.php and addActivityForm.php

4. Story 4: As a parent/guardian, I should be able to log a child's diaper changes so that I can track patterns to detect any problems.

Associated Scripts: index.php, addActivity.php and addActivityForm.php

5. Story 5: As a parent/guardian, I should be able to log a child's medicine intake (type, quantity, time) so that I can track their consumption.

Associated Scripts: index.php, addActivity.php and addActivityForm.php

6. Story 6: As a parent/guardian, I should be able to log a child's playtime so that I

can gauge how much physical activity they get.

Associated Scripts: index.php, addActivity.php and addActivityForm.php

7. Story 7: As a parent/guardian, I should be able to receive weekly and monthly reports so that I can analyze the data.

Associated Scripts: index.php, report.php and reportForm.php

### **User Stories for [Doctors]**

8. Story 8: As a doctor, I should be able to view a child's profile and account so that I can analyze their data and progress.

Associated Scripts: report.php, reportForm.php

9. Story 9: As a doctor, I should be able to view a child's weekly and monthly report so that I can assess the child's growth.

Associated Scripts: report.php, reportForm.php

### **• Forms and data handling work without errors.**

The forms all work without errors. The forms send data using POST and checks all the attributes. Some examples:

#### **updateActivities.php:**

Data Handling/Receiving the values using POST:

```
$activityID = filter_input(INPUT_POST, 'activityID', FILTER_VALIDATE_INT);
$categoryID = filter_input(INPUT_POST, 'categoryID',
    FILTER_VALIDATE_INT);
$babyID = filter_input(INPUT_POST, 'babyID', FILTER_VALIDATE_INT);
$userID = filter_input(INPUT_POST, 'userID', FILTER_VALIDATE_INT);
$time = filter_input(INPUT_POST, 'time');
getDescription = filter_input(INPUT_POST, 'description',
    FILTER_SANITIZE_STRING);
```

Validating the inputs:

```
if ($activityID === null || $activityID === false ||
    $babyID === null || $babyID === false ||
    $userID === null || $userID === false ||
    $time === null || $time === false) {
```

```

        //error message if anything is wrong
$error = "Invalid product data. Check all fields and try again.";
echo $error;
}

```

### **updateActivitiesForm.php:**

Validating:

```

if($activityID == null || $activityID === false){
    $error = "Error";
    echo $error;
    exit();
}

```

### **addActivity.php:**

Data Handling/Fetching inputs using POST:

```

$categoryID = filter_input(INPUT_POST, 'categoryID',
FILTER_VALIDATE_INT);
$babyID = filter_input(INPUT_POST, 'babyID', FILTER_VALIDATE_INT);
$userID = filter_input(INPUT_POST, 'userID', FILTER_VALIDATE_INT);
getDescription = filter_input(INPUT_POST, 'description',
FILTER_SANITIZE_STRING);
//this gets the date and time in the sql format
$time = date("Y-m-d H:i:s");

```

Validation:

```

if ($babyID === null || $babyID === false ||
    $userID === null || $userID === false) {
    $error = "Invalid product data. Check all fields and try again.";
    echo $error;
}

```

### **• Clear, readable, commented PHP code.**

See the PHP code for all comments. Some examples of comments throughout the code include:

**PHP comments:**

```

//start session and connect to database
session_start();
require('../inc/db_connect.php');

```

```

//fetch inputs
$categoryID = filter_input(INPUT_POST, 'categoryID', FILTER_VALIDATE_INT);

//insert activities into the database
$query = 'INSERT INTO activities
          (categoryID, babyID, userID, time, description)
          VALUES
          (:categoryID, :babyID, :userID, :time, :description)';

//back to the activity list index
include('index.php');

// hash password
$passwordHash = password_hash($password, PASSWORD_DEFAULT);

HTML comments:
<!-- Registration Form -->
<form method="post">
  <label>Full Name:</label>
  <input type="text" name="fullName"><br>

  <label>Username:</label>
  <input type="text" name="username"><br>

  <label>Password:</label>
  <input type="password" name="password"><br>

  <input type="submit" value="Register">
</form>

<!-- Link for registering -->
<p>Are you a doctor? <a href="registerDoctor.php">Register here</a></p>

<!-- input fields -->
<label>Baby ID:</label>
<input type="text" name="babyID"><br>

<!-- dropdown:-->
<label>Activity:</label>
<select name="categoryID">
  <?php foreach ($categories as $category) : ?>
    <option value="<?php echo $category['categoryID']; ?>">
      <?php echo $category['categoryName']; ?>

```

```
</option>
<?php endforeach; ?>
</select><br>

<!-- the head section -->
<head>
    <title>Kazeem's Kloud</title>
    <link rel="stylesheet" type="text/css" href="main.css">
</head>
```