Step 1: Set Up a Database

Start by creating a MySQL database to store job vacancy data. You'll need a table to store job details. Here's a simple SQL schema:

sql

CREATE TABLE job\_vacancies (

id INT AUTO\_INCREMENT PRIMARY KEY,

job\_title VARCHAR(255) NOT NULL,

job\_description TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

Step 2: Create the HTML Form

You've already started with the HTML form for adding job vacancies. You can enhance it by adding more input fields for additional job details like flyer, date, due date, division, location, and company name.

Step 3: Create PHP Scripts

Create PHP scripts to handle the form submission and interact with the database.

Insert Job Vacancy: When the user submits the form, use PHP to insert the job vacancy data into the database.

php

Copy code

<?php

// Connect to the database (use your own credentials)

$conn = new mysqli("localhost", "username", "password", "your\_database");

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$jobTitle = $\_POST["jobTitle"];

$jobDescription = $\_POST["jobDescription"];

$sql = "INSERT INTO job\_vacancies (job\_title, job\_description) VALUES ('$jobTitle', '$jobDescription')";

if ($conn->query($sql) === TRUE) {

echo "Job vacancy added successfully!";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

}

?>

List Job Vacancies: Create a PHP script to retrieve and display a list of job vacancies.

php

<?php

// Connect to the database

$conn = new mysqli("localhost", "username", "password", "your\_database");

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT id, job\_title, created\_at FROM job\_vacancies ORDER BY created\_at DESC";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "ID: " . $row["id"] . "<br>";

echo "Job Title: " . $row["job\_title"] . "<br>";

echo "Created At: " . $row["created\_at"] . "<br>";

// Add more fields as needed

echo "<hr>";

}

} else {

echo "No job vacancies found.";

}

$conn->close();

?>

Step 4: Display the List of Job Vacancies

In your HTML, create a section to display the list of job vacancies retrieved from the database.

Step 5: Styling and Validation

Add CSS styles to make your forms and listings visually appealing. Implement form validation to ensure that the user enters valid data.

Step 6: Update and Delete

To implement the "Update" and "Delete" functionalities, create additional PHP scripts and forms to edit and delete job vacancies.

Remember to handle security considerations, such as SQL injection prevention and user authentication, to make your CRUD system robust and secure.

This is a basic example to get you started with creating your own job vacancy management system. You can expand upon this foundation to include additional features and functionalities as needed.