Explanation of Approach:

The solution includes the following main components:

1. **Database Schema:**
   The schema includes three tables: employees, departments, and contacts. The relationships between them are:
   - An employee belongs to a department: The employees table has a department_id foreign key referencing the departments table.
   - An employee has a contact record: The contacts table references the employees table via the employee_id foreign key.
2. **Frontend Form :**
   - The form collects employee details like Name, Email, Department, and Phone Number.
   - On form submission, the data is sent to the server via AJAX (without refreshing the page).
   - A loader animation is displayed during the form submission, and a success or error message is displayed once the process is complete.
3. **AJAX Handler:**
   - The PHP script receives the form data, validates it, and processes the data (inserting into the database).
   - It sends a JSON response back to the frontend, indicating whether the operation was successful or if there was an error.
   - Errors (e.g., empty fields, database failures) are gracefully handled and returned as meaningful error messages.
4. **Database Utility Functions (File 3):**

   A set of reusable functions for database operations is provided. These functions:
   - Insert a new employee: Adds an employee's details to the employees table.
   - Associate the employee with a department: The department_id is inserted into the employees table to establish the relationship.
   - Save the employee’s contact details: Inserts the contact information (phone number) into the contacts table, associating it with the employee_id.
   - Retrieve data using SQL joins: Joins are used to demonstrate how to fetch employee details along with department and contact information.

Data Flow, AJAX Communication, and Error Handling:

1. **Data Flow:**
   - The user submits the form with employee details.
   - The data is sent to the server via AJAX (POST request).

- The PHP script receives the data, validates it, and inserts it into the appropriate tables in the database (employees and contacts).
- A JSON response is sent back to the frontend indicating the success or failure of the operation.

2. **AJAX Communication:**

    - The frontend sends the form data to the server using the $.ajax() method.
    - Upon receiving the data, the server processes it and returns a response in JSON format.
    - The frontend displays a loader while the request is being processed, and once completed, it shows either a success or error message.

3. **Error Handling:**

    - The script validates the incoming data. If any fields are missing, an error message is sent back to the frontend.
    - If a database operation fails, an exception is caught, and a meaningful error message is returned.
    - Frontend error handling ensures that appropriate messages are displayed for the user without page reloads.