**1. Datenbankdesign finalisieren**

* **Überprüfung und Korrektur des Datenbankdiagramms**: Stellen Sie sicher, dass alle Tabellen und Beziehungen korrekt sind. Berücksichtigen Sie die Beziehungen zwischen den Tabellen (z.B. Mitarbeiter und Abteilungen).
* **Normalisierung**: Überprüfen Sie, ob die Datenbank normalisiert ist, um Redundanzen zu vermeiden.
* **Indexierung**: Identifizieren Sie Schlüsselspalten, die indiziert werden sollten, um die Abfrageleistung zu verbessern.

**2. Datenbank erstellen**

* **SQL-Skripte schreiben**: Erstellen Sie SQL-Skripte, um die Datenbank und die Tabellen zu erstellen.
* **Datenbank in MySQL implementieren**: Führen Sie die Skripte in MySQL aus, um die Datenbank zu erstellen.

**3. Testdaten generieren**

* **Testdaten erstellen**: Erstellen Sie realistische Testdaten für jede Tabelle. Sie können Tools wie Mockaroo verwenden, um Testdaten zu generieren.
* **Testdaten in die Datenbank einfügen**: Verwenden Sie SQL-Insert-Befehle, um die Testdaten in die Datenbank zu laden.

**4. Backend-Entwicklung**

* **API erstellen**: Entwickeln Sie eine RESTful API, um auf die Datenbank zuzugreifen. Verwenden Sie ein Framework wie Flask (Python) oder Express (Node.js).
* **Endpoints definieren**: Erstellen Sie Endpoints für die Abfrage von Mitarbeiterdaten, PDF-Daten und Abteilungsinformationen.
* **Datenbankanbindung**: Implementieren Sie die Logik, um die API mit der MySQL-Datenbank zu verbinden.

**5. Frontend-Entwicklung**

* **Benutzeroberfläche erstellen**: Entwickeln Sie eine einfache Benutzeroberfläche, um mit dem Chatbot zu interagieren. Dies kann eine Webanwendung sein.
* **Chatbot-Integration**: Integrieren Sie den Chatbot in die Benutzeroberfläche. Sie können bestehende Chatbot-Frameworks wie Rasa oder Dialogflow verwenden.

**6. AI-Chatbot-Integration**

* **Chatbot-Training**: Trainieren Sie den Chatbot, um Benutzeranfragen zu verstehen und die entsprechenden Daten abzurufen.
* **API-Anbindung**: Stellen Sie sicher, dass der Chatbot in der Lage ist, die API-Endpoints aufzurufen, um Daten abzurufen.

**7. PDF-Datenverarbeitung**

* **PDF-Parsing**: Implementieren Sie eine Funktion, um PDF-Dateien zu parsen und relevante Informationen zu extrahieren. Sie können Bibliotheken wie PyPDF2 (Python) verwenden.
* **Datenintegration**: Integrieren Sie die extrahierten Daten in die Datenbank oder stellen Sie sie über die API bereit.

**8. Testing und Validierung**

* **Unit-Tests**: Schreiben Sie Unit-Tests für die API und die Datenbankabfragen.
* **Integrationstests**: Testen Sie die gesamte Anwendung, um sicherzustellen, dass alle Komponenten korrekt zusammenarbeiten.
* **Benutzertests**: Führen Sie Benutzertests durch, um die Benutzerfreundlichkeit und die Funktionalität des Chatbots zu überprüfen.

**9. Dokumentation**

* **API-Dokumentation**: Erstellen Sie eine Dokumentation für die API-Endpoints.
* **Benutzerhandbuch**: Erstellen Sie ein Benutzerhandbuch, das erklärt, wie man die Anwendung und den Chatbot verwendet.
* **README aktualisieren**: Aktualisieren Sie das GitHub-README mit detaillierten Informationen zum Projekt.

**10. Deployment**

* **Server einrichten**: Richten Sie einen Server ein, um die Anwendung zu hosten.
* **Anwendung deployen**: Stellen Sie die Anwendung auf dem Server bereit.
* **Monitoring einrichten**: Richten Sie Monitoring-Tools ein, um die Leistung und Verfügbarkeit der Anwendung zu überwachen.

**11. Wartung und Updates**

* **Fehlerbehebung**: Beheben Sie auftretende Fehler und Probleme.
* **Updates**: Implementieren Sie regelmäßige Updates, um neue Funktionen hinzuzufügen und die Sicherheit zu verbessern.