# amblem, simge, sembol, logo, ticari marka içeren bir resim Açıklama otomatik olarak oluşturuldu

**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CENG 301 LAB Project**

**Project Title:**

Human Resources Management Application

**Project Member(s):**

Fevzi FİDAN

Hikmet ÇATAK

Ömer Faruk AZİLİ

**Project Advisor**

Ahmet Serkan KARATAŞ, Ayşe Beyza ÜNAL

**17.11.2024**

Table of Content

* 1. **Project Introduction  
     1.1 Project Overview**

Briefly describe the purpose and objectives of the database project.

**1.2 Project Team**

**Roles and Responsibilities:**

Fevzi FİDAN: Responsible for creating the user interface and ensuring its usability.

Hikmet Çatak,Ömer Faruk Azili: Responsible for designing the database.

Hikmet Çatak:Responsible for writing Python functions that will run in app.(line=0-900 in db\_man\_projectv3 file)

Ömer Faruk Azili: Responsible for writing Python functions that will run in app.(line=900-1800 in db\_man\_projectv3 file)

**Project Scope**

Clearly define the scope of the database project, including its intended functionalities and limitations.

* 1. **Database Design  
     2.1 Introduction**

Provide an introduction to the database design, highlighting its purpose and significance in the context of the project.

**2.2 Entity-Relationship Diagram (ERD)**

metin, ekran görüntüsü, paralel, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

* 1. **Data Dictionary**

**Table:employees****:** Represents the employees in the company.  
  
**Columns:**

|  |  |
| --- | --- |
| **employee\_id** | int AI PK |
| first\_name | varchar(50) |
| last\_name | varchar(50) |
| date\_of\_birth | date |
| gender | varchar(50) |
| job\_title | varchar(50) |
| department | varchar(25) |
| salary | decimal(8,2) |
| hire\_date | date |
| email | varchar(50) |
| phone\_number | varchar(50) |
| password | varchar(50) |
| is\_active | tinyint(1) |

**Table:email: :** Represents the email records that sent.  
  
**Columns:**

|  |  |
| --- | --- |
| **id** | int AI PK |
| email\_title | varchar (255) |
| email\_description | text |
| **from\_emp\_id** | int |
| **to\_emp\_id** | int |

**-****An employee can send and get multiple email (many to many)**

**-From\_emp\_id, to\_****emp\_id** in the **email** is a **foreign key** that references the **employee\_id** column in the **employees** table.

**Table:employee\_items:** Represents the assigned items to employees.  
  
**Columns:**

|  |  |
| --- | --- |
| **employee\_id** | int |
| **item\_id** | int |
| assignment\_date | datetime |
| quantity | int |
| **assign\_id** | int AI PK |

**-** **More than one item can be assigned to more than one employee based on quantity. (many to many)**

**-employee\_id** in the **employee\_items** is a **foreign key** that references the **employee\_id** column in the **employees** table.

**Table:employee\_leaves:-Ö**  
  
**Columns:**

|  |  |
| --- | --- |
| **leave\_request\_id** | int AI PK |
| **employee\_id** | int |
| status\_of\_request | enum('Pending','Accepted','Rejected') |
| request\_date | date |
| approved\_by | int |
| answer\_date | date |
| leave\_type | enum('Annual Leave','Health Leave','Unpaid Leave','Excuse Leave') |
| Start\_date | date |
| end\_date | date |
| total\_dates | int |
| desc\_request | text |
| created\_at | timestamp |

**-employee\_id** in the **employee\_leaves** is a **foreign key** that references the **employee\_id** column in the **employees** table.

**Table:events\_:Represent the events.**  
  
**Columns:**

|  |  |
| --- | --- |
| **id** | int AI PK |
| event\_name | varchar(255) |
| event\_text | text |
| event\_date | datetime |

**Table:items:Represent the items in company.**  
  
**Columns:**

|  |  |
| --- | --- |
| **id** | int AI PK |
| item\_name | varchar(20) |
| quantity | int |

**Table:messages:Represent the messages informations.**  
  
**Columns:**

|  |  |
| --- | --- |
| **id** | int AI PK |
| **from\_emp\_id** | int |
| **to\_emp\_id** | int |
| message\_text | text |
| is\_read | tinyint(1) |
| message\_date | datetime |
| subject | varchar(255) |

- **An employee can send and get multiple messages (many to many)**

**-****From\_emp\_id,to\_emp\_id** in the **messages** is a **foreign key** that references the **employee\_id** column in the **employees** table.

**Table:pending\_email:Represent the emails that will be send .**  
  
**Columns:**

|  |  |
| --- | --- |
| **id** | int AI PK |
| email\_title | varchar(255) |
| email\_description | text |
| **from\_emp\_id** | int |
| **to\_emp\_id** | int |
| send\_date | datetime |
| is\_sent | tinyint(1) |

**-From\_emp\_id,to\_emp\_id** in the **pending\_email** is a **foreign key** that references the **employee\_id** column in the **employees** table.

**Table:special\_requests:Ö-**  
  
**Columns:**

|  |  |
| --- | --- |
| **request\_id** | int AI PK |
| **employee\_id** | int |
| request\_type | enum('Advance','Salary Increase','Payback','Other') |
| request\_amount | decimal(10,2) |
| request\_date | date |
| status\_of\_request | enum('Pending','Accepted','Rejected') |
| approved\_by | int |
| description | text |
| answer\_date | date |
| created\_at | timestamp |

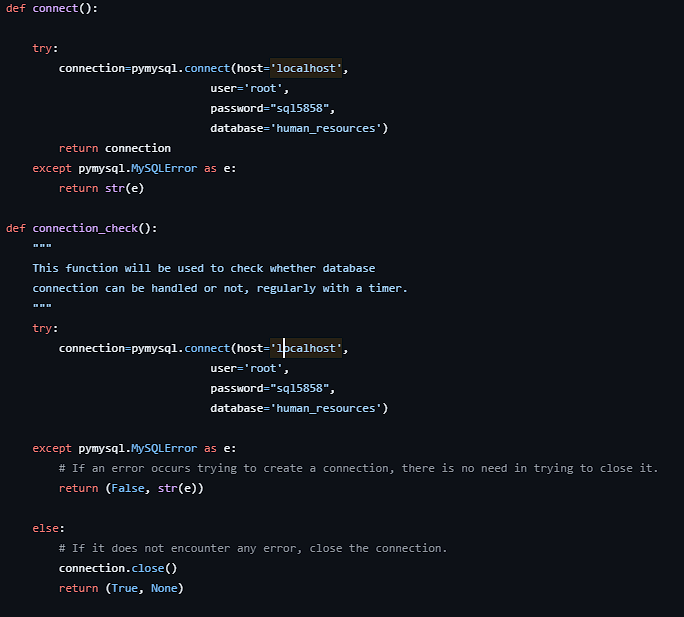
**View:employee\_items\_with\_names:** **Shows employees who have items assigned to them.**  
**Columns:**

|  |  |
| --- | --- |
| employee\_id | int |
| first\_name | varchar(50) |
| last\_name | varchar(50) |
| item\_id | int |
| item\_name | varchar(20) |
| quantity | int |
| assign\_id | int |
| assignment\_date | datetime |

* 1. **Database Implementation**  
     3.1 **Database Management System**

MySQL

* 1. **Database Connectivity**



We used pymysql library to connect database and implementing functions in app.

**Parameters:**

Host= Specifies the database server's hostname.

User= The username used to connect to the database.

Password= The password associated with the username **root** to authenticate the connection.

Database= Specifies the name of the database.

If the connection is successful, the connection object is returned. This object allows you to interact with the database (e.g., execute queries, fetch results).

* 1. **Interface Design**  
     5.1 **User Interface Design**

The interface of the project has been developed in Python. We used PyQt6 GUI toolkit.

Specify the programming languages and tools used to create the interface (e.g., Java, C#, PHP, HTML, CSS, etc.).

* 1. **Interface Overview**

Provide screenshots of each interface page, explaining its purpose and functionality. ( Examples: Login Page, Product Management Page (e.g., Add Product, Delete Product), Customer Information Page, Navigation and Buttons, ect…)

* 1. **Appendices**
* Include all relevant code used for the project, organized by functionality (e.g., database schema, connectivity code, interface design scripts).

---------------------------------------------------------------------------------------------------------------

* 1. **System Testing and Validation**

**(No writing here—this section is for live demonstration!!)**

* After presenting all components, test the program in front of your classmates.
* Demonstrate functionalities like adding, deleting, and modifying records through the interface and confirm that the changes are reflected in the database.