HOSPITAL MANAGEMENT SYSTEM

Patient Tracking and Management System: This project is a web application designed as a hospital management system. This application is expected to provide a platform where patients can create records, make appointments with doctors, store medical reports and generally manage health related transactions.

Database Design: Below are the basic tables that should be included in the database. You can create new tables and relationships between these tables during the project development phase. The database should be made relational by determining the *primary key* and *foreign key* for each table. You are expected to create your tables *according to normalization rules* (1NF, 2NF and 3NF etc.).

1. **Patients:**PatientID, FirstName, LastName, Date of Birth, Gender, PhoneNumber, Address, etc.
2. **Doctors:**DoctorID,FirstName,LastName,SpecialtyArea,HospitalOccupation.
3. **Admin:**AdminIDetc.
    a.An administrator can add a patient.
    b.A manager can add a doctor.
    c.A manager can add a medical report.
4. **Appointments:**AppointmentID,AppointmentDate,AppointmentTime, etc.
5. **MedicalReports:**ReportID,ReportDate,ReportContent, etc.
    a.Medical reports are image files.
    b.*Storing Image Files*: To store lab results as images, you must upload the files to a file storage system and store URLs pointing to these files in the database. These URLs should be added as a column in the Lab Results table.
    c.Medical reports can be attached by the patient, doctor or administrator.

Object Oriented Programming During the development of this project it is imperative to use the principles of object- oriented programming. Each component (patient, doctor, appointment, medical report, etc.) is expected to be modeled as a class and appropriate methods are expected to be defined for each class. It is expected that trigger functions will be written for insertions or deletions made to a table and all other related tables will be updated.

Safety Precautions: HTTPS protocols or appropriate encryption methods should be used to secure sensitive data such as lab results.

Interface Enhancements:
All changes made in the database should be tracked through the interface.

1.Renewing the User Interface: An interface where patients and doctors can view and upload their appointments and laboratory results should be added. Organize this interface to show content information such as dates and types of results.

2. Using Dynamic Components: Use AJAX calls in the interface to upload files without refreshing the page. Dashboards should be added to patient and doctor profiles so that users can see their medical history, treatment notes and appointment history.

3.Queries: The results of all queries should be viewable through the interface. For example, a doctor should be able to view all his/her patients and the medical report results of a patient, or a patient should be able to view all his/her appointments and medical report results;

a.A doctor should be able to view all his/her patients, listing basic information such as each patient's name, surname, date of birth, gender, phone number, address.

b.A doctor should be able to view the results of medical patient. Information such as the date, content, reports for a specific results of the reports should be presented along with the patient's name and surname.

c.A patient should be able to view all their appointments. Information such as the date, time and doctor of each appointment should be listed.

d.A patient should be able to view all medical report results. Information such as date, content, results of reports should be presented.

Project Requirements:

1.Patient Addition: A new patient entry to the database should be possible through the interface and updates in all relevant tables should be shown in the database during the project presentation.

2.Add Doctor: Enter a new doctor in the database through the interface and updates to all relevant tables should be shown in the database during the project presentation.

3.Making an Appointment It should be shown whether a patient has made an appointment with a specific doctor and whether this appointment has been correctly updated in the database.

4.Attach Medical Report: Attach a patient's medical report to show whether the report has been correctly saved in the database.

5.Update Patient Information: Update the information of an existing patient via the interface update to show whether this change has been correctly updated in the database.

6.Update Doctor Information: Update an existing doctor's information through the interface to show whether this change has been correctly updated in the database.

NOTE: When the operations specified in the requirements are executed, a confirmation message should be displayed on the interface and the database changes should be reported instantly.

Assignment Submission

- Recommended Programming Languages: Python, PHP, JavaScript (Node.js).
- Recommended DBMS: MSSQL, PostgreSQL, MySQL, SQLite.
- The project report should be 4 pages long in IEEE format. The report should include a flow diagram or pseudocode, abstract, introduction, methodology, experimental results, conclusion, references and ER diagram. This is a database project and does not have reports will not be considered!
  an ER diagram.
- The follow-up of the course will be done through the edestek2.kocaeli.edu.tr system, including the delivery of the project. Projects submitted after the date specified on edestek2.kocaeli.edu.tr will not be accepted. Questions about the project will be answered through the forum on edestek2.kocaeli.edu.tr
- You may be asked questions about the algorithm, the purpose of the various parts of the code you are developing and the development environment.
- You may be asked questions about the tables you created in the database and normalization processes.
- You may be asked to describe any line code you use.