# Clinic Management System

A small clinic wants to allow their front office staff to manage the visiting doctor’s appointment though a web application.

Users in this document refer to the front office staff. The system provided the below capabilities,

1. Setup Patient information
2. Setup Doctor information
3. Fix an appointment for patient with a doctor
4. Cancel appointment

The system will be secured that only logged in users can perform any of the above functions.

## 

## Detailed Requirements

### User Setup

Each Front office staff will have a unique user name and password. There is no screen to manage the users, the data will be managed in the user table directly in the database. The database table will have the below data for all the front office staff.

1. User name (unique, no special characters)
2. First Name
3. Last Name
4. Password

### Screen 1: Login

This screen will prompt the user to enter their user name and password. Both fields are mandatory and to be validated. The page will also have a small clinic logo (use the logo from <https://commons.wikimedia.org/wiki/File:Stone-Clinic-Logo.png> )

The user name and password will be validated at the server side against the database table – if the details match, the users will be redirected to the Home page and if not, they will get the same login page with the error message “You’ve entered an incorrect user name or password”.

### Screen 2: Home

This screen will be shown only when the user has successfully logged in. This will have the clinic logo (the one used in the login screen) and also a navigation bar which will have the below options,

|  |  |  |
| --- | --- | --- |
| **S.No** | **Menu Name** | **Comments** |
| 1 | Add Doctor | This will redirect to the “Add Doctor” screen |
| 2 | Add Patient | This will redirect to the “Add Patient” screen |
| 3 | Schedule Appointment | This will redirect to the “Schedule Appointment” screen |
| 4 | Cancel Appointment | This will redirect to the “Cancel Appointment” |
| 5 | Logout | This will log off the user and redirect to the “Login” screen |

### Screen 3: Add Doctor

Users will use this screen to add the Doctor information to the system. Screen will have the below fields,

1. Doctor Id (auto generated number by the system, not editable by the users)
2. First Name (No special characters allowed)
3. Last Name (No special characters allowed)
4. Sex (Choices - M/F/Others)
5. Specialization (Choices – General, Internal Medicine, Pediatrics, Orthopedics, Ophthalmology)
6. Visiting Hours (Doctors can visit for 1hr, 2hrs, 3hrs etc., and not 1:30hrs or 1:15 or 1:45hrs)
   1. From – HH:MM
   2. To – HH:MM

When the users submit the data, validate and save to the database.

### Screen 4: Add Patient

Users will use this screen to add Patient information to the system. Screen will have the below fields,

1. Patient Id (auto generated number by the system, not editable by the users)
2. First Name (No special characters allowed)
3. Last Name (No special characters allowed)
4. Sex (Choices - M/F/Others)
5. Age (0 to 120 only allowed)
6. Date of birth (DD/MM/YYYY – Indian format)

When the users submit the data, validate and save to the database.

### Screen 5: Schedule Appointment

Users will use this screen to setup the Patient-Doctor appointments. Each appointment is for one hour. Screen will have the below fields,

1. Patient Id
2. Specialization Required (Choices – General, Internal Medicine, Pediatrics, Orthopedics, Ophthalmology)
3. Doctor (System will display the Doctors based on the Specialization chosen above)
4. Visit Date (DD/MM/YYYY – Indian format)
5. Appointment Time (Show the specific time slots available for the doctor for the given date, each slot is 1 hour)

When the users submit the data, validate and save to the database. Each appointment will have a unique id in the database.

### Screen 5: Cancel Appointment

Users will use this screen to cancel the Patient-Doctor appointments. Each appointment is for one hour. Screen will have the below fields,

1. Patient Id
2. Visit Date (DD/MM/YYYY – Indian format)
3. Appointment List (Based on the patient id and the visit date, the system will display the appointments for the patient)

User can choose only one appointment and then hit the “Cancel Appointment“ button to proceed to cancel the appointment.

When an appointment is cancelled, the slot will become available for other patients.

### General Requirements

* All screens except “Login” and “Home” will have a button “Go to Home” to take the user back to the Home screen
* Whenever an operation is successful, the users should be shown a message as such. E.g., if the “Add Patient” is successful, it will display “Patient successfully added to the system”. It would be helpful to show the names/ids in these messages as well to make it usable.
* On failures, the users should be shown appropriate friendly error message (not exceptions).

### Technical Requirements

* Tables should be designed appropriately (proper data types, table/column names, constraints), use a SQL Server database
* The tables should be created only using SQL Files and these should be available in the repo
* Front end – ASP.NET Core MVC, HTML5, CSS, Bootstrap, Javascript/Jquery
* Business Logic – Web API/Service in MVC
* Data Access – EF Core (DB/Model first approach)
* All names used in the code (classes, variables, properties etc.,) should be meaningful and follow consistent naming approach (Camel Case, Pascal Case etc.,)
* Classes/methods/properties should have proper accessibility modifiers
* Use code formatting to ensure the code is readable, write comments where required so that the reviewers can understand your code

### IMPORTANT

* Create a new repo “ProjectA” in your account and use that for this exercise
* Push the code to the remote repository often or at least before end of each day.
* Repo should have a folder “DB Script” and it will have all the SQL Files (tables, stored procedure etc.,)
* Repo should have a folder “src” and it will have your ASP.NET Core Projects