

1. Angular 9 - Healthcare Service

Healthcare Service

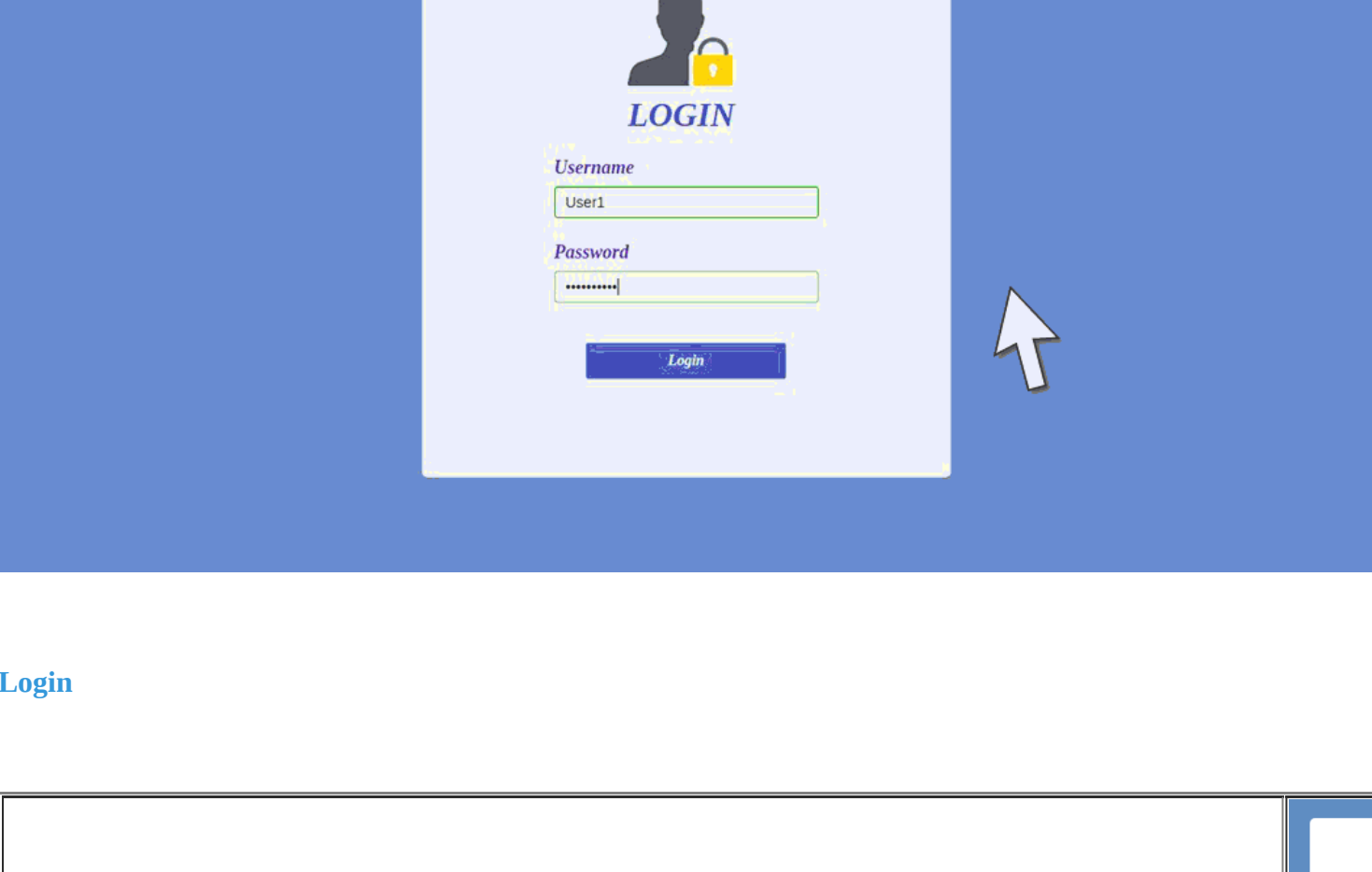
Create an Angular application **Healthcare Service** that can be used by the receptionist of a hospital to register patients and book/cancel appointments.

Application Overview

The following are the screens in the application:

- Login
- Profile
- Register patient
- Viewall Patients
- Requested App

Application Workflow



Login

<ul style="list-style-type: none">• The LOGIN page must contain Username and Password fields, and a Login button.• If there are no inputs in the Username/Password fields, an error message must be displayed.• The Login button should be enabled only if all the fields pass the required validations.	
<ul style="list-style-type: none">• The Username should be a minimum of 3 characters, and the Password should be a minimum of 8 characters.• Password Pattern: Should contain at least one upper case letter, one lower case letter, one numeric digit, and one special character. The allowed special characters are !\$%&#EC*'?&. Example: pas\$word8#• Error messages must be displayed for invalid inputs.	
<ul style="list-style-type: none">• The Username and Password should not exceed 20 characters.• Error messages must be displayed for invalid inputs.	

Profile

<ul style="list-style-type: none">• The Profile page must contain a user profile section (Name, Contact number, Email, and Location) with an Edit profile button.• A user should be able to edit their profile details in the form that appears when the Edit profile button is clicked.	
<ul style="list-style-type: none">• A user should be able to edit all details except Name (disabled).• The Contact number should be exactly 10 digits, and Email should be a valid email ID. Error messages must be displayed for invalid inputs.• When the user successfully submits the updates, the changes must be displayed.• When the Discard button is clicked, the form must be hidden.	
<ul style="list-style-type: none">• None of the input fields should be empty.• The Make changes button should be enabled only if all the fields pass the required validations.	

Register Patient

<ul style="list-style-type: none">• The fields marked * are mandatory.• The Submit button should be enabled only after the user has entered details in all the mandatory fields.• On submitting the registration form, the user should be redirected to the View all patients page where the list of all the patients should be displayed.	
<ul style="list-style-type: none">• First Name should be a minimum of 3 characters and a maximum of 20 characters, and should not be empty.• Last Name should be a minimum of 3 characters and a maximum of 20 characters, and should not be empty.• Gender should not be empty.• DOB should not be empty.• The Contact number should be exactly 10 digits and should not be empty.• The Email should be valid.• The description field is optional.• Error messages must be displayed for invalid inputs.	

View all Patients

<ul style="list-style-type: none">• Should display all the registered patients.	
<ul style="list-style-type: none">• On clicking View Details, details of the selected patient should be displayed.	
<ul style="list-style-type: none">• On clicking the Book Appointment button, the Book Appointment form should be displayed.• The fields marked with * are mandatory.• The Schedule button should be enabled only after the user has entered details in all the mandatory fields.• After booking an appointment, the user should be redirected to the Requested Appointments page where all the appointments should be displayed.	
<ul style="list-style-type: none">• The Scheduled Appointments table should be displayed when the Scheduled Appointments button is clicked.• The table should display all the appointments of the selected patient.• On clicking the Cancel Appointment button, the appointment should be deleted.	
<ul style="list-style-type: none">• Display an appropriate message if there is no appointment.	

Requested Appointments

<ul style="list-style-type: none">• Should display the booked appointments of all the patients.• On clicking View Details, details of the selected patient should be displayed.• On clicking the Cancel Appointment button, the appointment should be deleted.	
--	--

Header

<ul style="list-style-type: none">• The header component should be displayed at the top of all pages except the Login page.• The header component should display the name of the logged-in user.• Clicking on a particular menu should navigate to the corresponding page.	
---	--

Services

Services in the application:

- **API Service**
Service for API calls to the backend server.
- **Auth Guard**
To prevent navigation to the application pages other than the login page for unauthorized users.
- **Data Service**
Acts as middleware between a component and API service.

Other Instructions

- All the project files should be kept in the **src** folder.
- **Do not modify the ids or classes** of the existing UI elements as the unit tests require them.
- The API URL is predefined in the service as **API_URL**. You can use **API_URL + endpoint** as required.
- To **log in**, use the credentials available at **server/users.json**.

Methods in Service	API Endpoint	HTTP Method	Request body	Response
checkLogin	/auth/server/	POST	{ username: username, password: password }	{userid: userId}
getUserDetails	/users/ userId	GET		user object
updateDetails	/users/ userId	PUT	user object	user object
registerPatient	/allpatients	POST	patientDetails	patientDetails
getAllPatientsList	/allpatients	GET		patientDetails
getParticularPatient	/allpatients/ patientId	GET		patientDetails
getDiseasesList	/diseases	GET		diseasesList
bookAppointment	/reqappointments	POST	appointmentDetails	appointmentDetails
getAppointments	/reqappointments?patientId= patientId	GET		appointmentDetails
deleteAppointment	/reqappointments/ appointmentId	DELETE		{}
requestedAppointments	/reqappointments	GET		appointmentDetails

- Notes:**
- Replace the colored text with the appropriate value.
 - If the backend server stops responding, rerun the application, and restart the server.