

1. RBP - Health Care Service - full stack

Health Care Service

Health Care Service is an Angular application used by receptionist of a hospital. This application helps the receptionist to register patient/book appointment/cancel appointment.

Nodejs:

You have to create an API service for Health Care Service.

Below are the details of the API endpoints.

These are the API endpoints

- login
- register
- addPatient
- fetchSinglePatient
- editProfile
- diseases
- bookAppointment
- fetchAppointment
- deleteAppointment
- singlePatientAppointment
- getProfile

Functionalities of endpoint are given below:

login:

In this endpoint, you have to validate the login details of the user with received parameters like uname, pwd and respond with JSON data object as below

Key	Value
status	boolean
message	string
token	string
uid	string
mobile	number
email	string
location	string
userName	string

Details about the token in the data is explained at the end of question in middleware section.

register:

In this endpoint, you will receive request parameters like email, pwd, uname, mobile,location.
you should get these parameters and insert into the database as an object which will be in the below structure.

Key	Value
_id	mongoose.Types.ObjectId()
email	string
pwd	string
userName	string
mobile	Number
location	string

Note: Storing password as plain text is not at all good practice, This is all about demonstration purpose to create an API.
The general practice could be hash the password and store in the database.

addPatient:

In this endpoint, you will receive request parameters like fname, lname, gender, dob, mobile, email, desc, userId
you should get these parameters and insert into the database as an object which will be in the below structure.

Key	Value
_id	mongoose.Types.ObjectId()
fname	string
lname	string
gender	string

Key	Value
dob	Date
mobile	Number
email	string
desc	string
userId	string

fetchSinglePatient:

In this endpoint, you will receive request parameter like patientId.

you should get this parameter and you should query the patient collections with received patientId and return the response which can be as below.

You use MongoDB as database and details about the required collections and their structures are the end of the question.

```
[
  {
    _id: mongoose.Schema.Types.ObjectId,
    fname: String,
    lname: String,
    gender:String,
    dob: Date,
    mobile:Number,
    email:String,
    desc:String,
    userId:String},
  {..},
  {..}
]
```

After querying , if no match with patientId is found in database, response can be an empty array too.

editProfile:

In this endpoint, you will receive request parameters like email, mobile, location, uid.

Fetch the object from user collection based on uid, edit the parameters of the object and save it in the database.

You should save the object as below

Key	Value
_id	mongoose.Types.ObjectId()
userName	string
pwd	string
mobile	Number
email	string
location	string

You need to edit only received parameter values only.

diseases:

Request method used in this API is GET.

In this endpoint, you have to query the diseases collections and return the JSON data obtained from MongoDB.

The diseases data that should be in the "diseases" collection is mentioned below:

```
{diseases:['flu','cold','fever','typhoid',.....]}
```

The array can contain any disease name you want.

bookAppointment:

In this endpoint, you will receive request parameters like fname, lname, disease, priority, tentativeDate, patientId, registeredTime.

You have to save these details in 'appointments' collection in MongoDB.

The JSON object that should be saved should be as below:

```
{
  _id: mongoose.Schema.Types.ObjectId,
  fname: String,
  lname: String,
  disease:String,
  priority:String,
  AppointmentDate:Date,
  patientId:String,
  bookingTime: Date
}
```

fetchAppointment:

In this endpoint, You have to GET all the appointments from the 'appointments' collections.

The response after querying 'appointments' collection can be as below:

```
[
  {
    _id: mongoose.Schema.Types.ObjectId,
    fname: String,
    lname: String,
    disease:String,
    priority:String,
    AppointmentDate:Date,
    patientId:String,
    bookingTime: Date
  },
  {..},
  {..},...
]
```

After querying , if there are no appointments, response can be an empty array too.

deleteAppointment:

In this endpoint, you will receive an request parameter like appointmentId,

You have to delete object with id equals to appointmentId.

On successful deletion, you have to return an JSON object with 'status' as key and "success" as value in response with status code as 200.

On deletion failed, you have to return an JSON object with 'status' as key and "failure" as value in response with status code 404.

NOTE: Every Object in MongoDB has an object id with key as '_id'.

In this API, received appointmentId is the `_id` of the object that you need to delete.

singlePatientAppointments:

If you remember, in the `fetchAppointment` API, you fetch all the appointments available in database.

But in this API you have to fetch a single patient appointment based on the `patientId` which we store while booking an appointment using `bookAppointment` API.

In this API, you receive request parameter like `patientId`.

You have to query the 'appointments' collection based on `patientId` and return the JSON response.

getProfile:

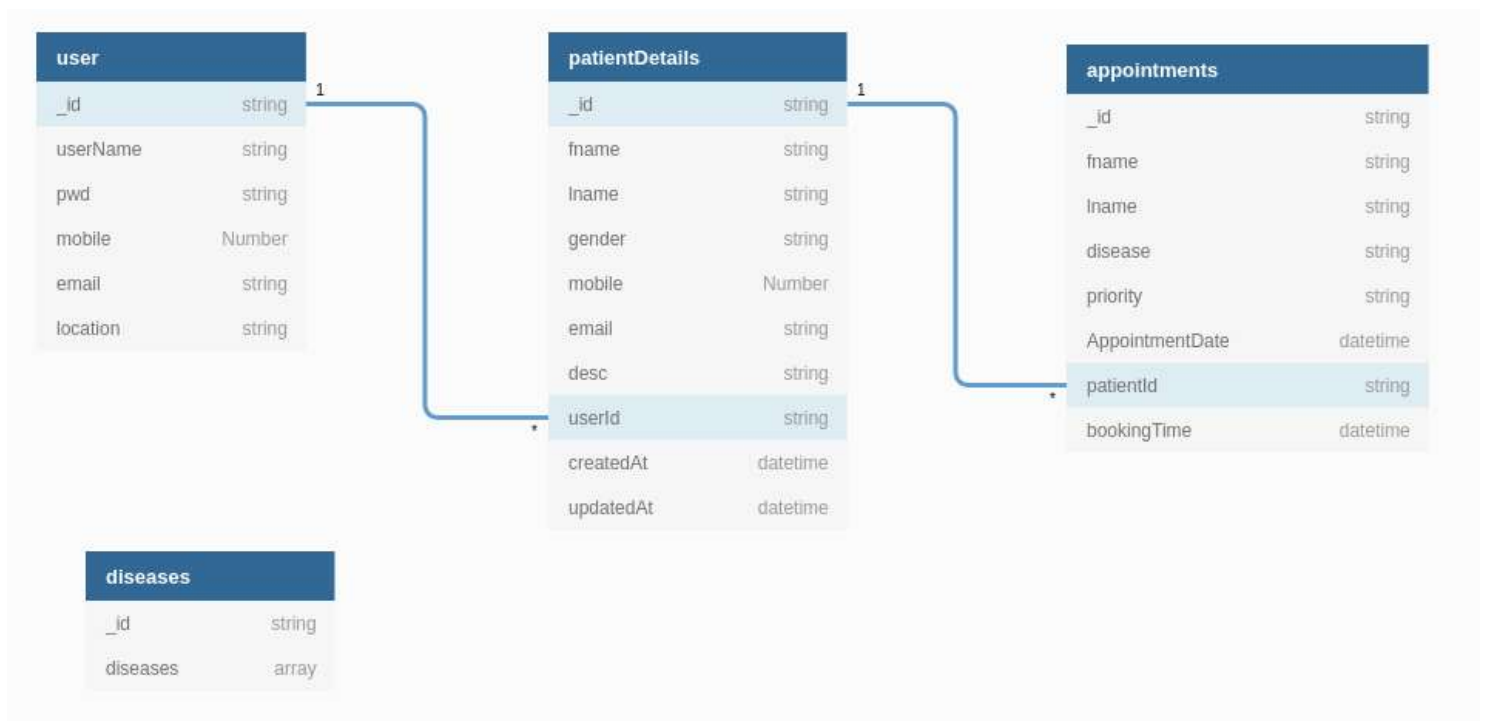
In this API, You have send the user details.

You receive request parameters like `uid`, with which you have to query 'user' collection and send the query results as JSON.

BUT the response object shouldn't have the 'pwd' property.

perform operation to delete the property from the response object.

Internal Structure of collection and meaning of attributes in MongoDB:



While structuring the database models, we consider below factor:

user:

_id	unique ID generated By MongoDB
userName	username of the user
pwd	Password of user
mobile	user's mobile number
email	email address of user
location	location of user

patientDetails:

_id	unique ID generated By MongoDB
fname	firstname of patient
lname	last name of patient
gender	gender of Patient
dob	Date of birth of patient
mobile	phone number of patient
email	email address of patient
desc	description provided by patient
userId	userId of user who added patient data

This model contains 'timestamps' as true, which create createAt and updatedAt field in objects which indicates the time at which object is created and updated.

appointments:

_id	unique ID generated By MongoDB
fname	first name of patient
lname	last name of patient
disease	name of the disease patient has
priority	Priority of patient
AppointmentDate	date of appointment
patientId	Id of patient
bookingTime	Time at which booking is made

diseases:

_id	unique ID generated By MongoDB
diseases	Array of diseases

IMPORTANT:**Middleware:**

You need to create an Middleware to validate the API calls.

For Middleware, you have to use 'json web token' (jwt) with below rules:

1) login and registration API endpoint don't need middleware. If login details are validated , you send response with json web token as token parameter.

2) With obtained token,for every request, the request should have 'x-access-token' or 'authorization' as key and token as value.

Note:

- **Run -> install :** To install all the node modules for both frontend and backend, to install MongoDB and insert dummy data into it
- **Run -> Run:** To see running application
- **Run -> Test:** To see results of testing the API and react behaviour as expected

Angular:

Application Overview

The application has the following screens


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

Signup

<ul style="list-style-type: none">• The form should contain the fields username, password, contact number, Email and location.• All input fields should not be empty. Create Account button should be enabled only if all fields pass required validations.• Contact number should be exactly 10 numbers and email should be a valid email. Show error message for invalid inputs.	<div><h3>SIGN UP</h3><p>Username</p><input type="text" value="username"/><p>You must enter a username</p><p>Password</p><input type="password" value="password"/><p>You must enter a password</p><p>Contact number</p><input type="text" value="mobile number"/><p>You must enter a valid mobile number</p><p>Email</p><input type="text" value="Email"/><p>You must enter a valid Email ID</p><p>Location</p><input type="text" value="Location"/><p>You must enter the location</p><div><button>Back</button><button>Create Account</button></div></div>
<ul style="list-style-type: none">• Username should be a minimum of 3 characters and password should be a minimum of 8 characters.• Password Pattern: should have at least one upper case letter, one lower case letter, and one numeric digit and one special character. The allowed special characters are !\$%#@#€€*?&• Example Password: pasSword8#• Show error message for invalid inputs.	<div><p>Username</p><input type="text" value="x"/><p>User name must be at least 3 characters long</p><p>Password</p><input type="password" value="•"/><p>Password must be at least 8 characters long Pattern does not match</p></div>

<ul style="list-style-type: none">Username and password should be a maximum of 20 characters.Show error message for invalid inputs.	<div><div>Username</div><div>XXXXXXXXXXXXXXXXXXXXX</div><div>Username cannot exceed 20 characters</div><div>Password</div><div>..... </div><div>Password cannot exceed 20 characters</div><div>Pattern does not match</div></div>
--	---

Login

<ul style="list-style-type: none">Login page has username and password fields and a login , signup button.If Username/Password input field is empty, an error message should be shown for empty username/password field.Login button should be enabled only if all fields pass required validations.	<div><div><div>LOGIN</div></div><div><div>Username</div><div>username</div><div>You must enter a username</div><div>Password</div><div>password</div><div>You must enter a password</div><div>Login</div><div>Sign up</div></div></div>
<ul style="list-style-type: none">Username should be a minimum of 3 characters and password should be a minimum of 8 characters.Password Pattern: should have at least one upper case letter, one lower case letter, and one numeric digit and one special character. The allowed special characters are !\$%#@#£€*?&Example Password: pasSword8#Show error message for invalid inputs.	<div><div>Username</div><div>x</div><div>User name must be at least 3 characters long</div><div>Password</div><div>• </div><div>Password must be at least 8 characters long</div><div>Pattern does not match</div></div>
<ul style="list-style-type: none">Username and password should be a maximum of 20 characters.Show error message for invalid inputs.	<div><div>Username</div><div>XXXXXXXXXXXXXXXXXXXXX</div><div>Username cannot exceed 20 characters</div><div>Password</div><div>..... </div><div>Password cannot exceed 20 characters</div><div>Pattern does not match</div></div>

Profile

- Profile page have a user profile section that has Username, Contact number, Email and Location with an edit profile button.
- User can update the details by clicking **Edit profile** button which displays a form where user can edit details

Health Care Service Welcome **User1** Profile Register patient Viewall patients Requested App Logout

User1
 9999999999
 user11@abc.in
 bangalore
Edit profile

- User should be able to change the details except for username which is disabled
- Contact number should be exactly **10 numbers** and email should be a **valid email**. Show error message for invalid inputs.
- Once user successfully submit the updates, the user should be able to view the changes on the screen
- should display user profile section and hide form when clicked on **discard** button

Health Care Service Welcome **User1** Profile Register patient Viewall patients Requested App Logout

Edit your profile
Name

Contact number

Email

Location

Make changes
Discard

- All input fields should not be empty. Make Changes button should be enabled only if all fields pass required validations.

Edit your profile

Name

Contact number

You must enter a valid mobile number

Email

You must enter a valid Email ID

Location

You must enter the location

Make changes
Discard

Register Patient

- * fields are mandatory.
- Submit button should be enabled once the user filled all the * fields.
- The submission of registration form should redirect the user to View all patients Page and the page should display the list of all patients.

Enter Patient details

First Name: *

FIRSTNAME

Last Name: *

LASTNAME

Gender: *

Male

Female

Date of birth: *

dd/mm/yyyy

Contact number: *

mobile number

Email: *

example:abc12@tcs.com

Description: (Optional)

Briefly describe the issue

Submit

- 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.
- Date of birth should not be empty.
- Contact number should be 10 numbers and length should not exceed 10 digits and should not be empty.
- Email should be a valid email.
- Description is optional
- Show error message for invalid inputs.

First Name: *

FIRSTNAME

You must include a first name.

First Name: *

X

Your first name must be at least 3 characters long.

First Name: *

xxxxxxxxxxxxxxxxxxxxxx

Your first name cannot exceed 20 characters.

View all patients

- Should display the all registered patients.

Health Care Service

Welcome User1

Profile

Register patient

Viewall patients

Requested App

Logout

Patient ID	FirstName	LastName	Gender	DOB(yyyy-MM-dd)	Mobile	Email	Registered on	Update
400	first	name	Female	2019-02-07	9998887775	first1@abc.com	11/02/2019 03:29:08	View Details
401	second	name	Male	2019-01-30	3545454555	second2@def.com	11/02/2019 03:40:44	View Details
402	third	name	Female	2019-09-13	4466162463	third3@ghi.com	16/09/2019 11:09:51	View Details
403	fourth	name	Male	2019-09-12	6666666666	fourth4@jkl.com	16/09/2019 11:16:50	View Details
404	fifth	name	Female	2019-09-13	9876543111	fivth5@mno.in	17/09/2019 12:37:27	View Details
405	sixth	name	Female	2019-10-17	8800990077	sixth6@abc.in	22/10/2019 02:54:57	View Details

- Should display selected patient details, when clicking on view details of particular patient

Health Care Service

Welcome User1

Profile

Register patient

Viewall patients

Patient Details

ID: 404

Name: fivth name

Gender: Female

Date of birth: 2019-09-13

Mobile: 9876543111

Email: fivth5@mno.in

Registered on: 17/09/2019 12:37:27

Book Appointment

Scheduled Appointment

<ul style="list-style-type: none">Should show book appointment form when clicked on book appointment button.* fields are mandatory.Schedule button should be enabled once the user filled all the * fields.After booking an appointment, should redirect to Requested appointments Page and the page should display all appointments.	<div><h3>Book Appointment</h3><div><div>Select Disease: *<div>Select disease</div></div><div>Priority: *<div>Select priority</div></div><div>tentativeDate: *<div><div>dd/mm/yyyy</div></div></div><div>Schedule</div></div></div>																		
<ul style="list-style-type: none">Should show scheduled appointment table when clicked on scheduled appointment button.Should display the all appointments of selected patient.Should delete appointment when clicked on cancel appointment button.	<div><h3>Scheduled Appointment</h3><table><thead><tr><th>Appointment id</th><th>Disease</th><th>Priority</th><th>Tentative date</th><th>Requested on</th><th>Update</th></tr></thead><tbody><tr><td>17</td><td>acid reflex</td><td>Normal</td><td>22/10/2019</td><td>22/10/2019 06:17:17</td><td>Cancel Appointment</td></tr><tr><td>18</td><td>fever, pain</td><td>Normal</td><td>23/10/2019</td><td>22/10/2019 06:18:09</td><td>Cancel Appointment</td></tr></tbody></table></div>	Appointment id	Disease	Priority	Tentative date	Requested on	Update	17	acid reflex	Normal	22/10/2019	22/10/2019 06:17:17	Cancel Appointment	18	fever, pain	Normal	23/10/2019	22/10/2019 06:18:09	Cancel Appointment
Appointment id	Disease	Priority	Tentative date	Requested on	Update														
17	acid reflex	Normal	22/10/2019	22/10/2019 06:17:17	Cancel Appointment														
18	fever, pain	Normal	23/10/2019	22/10/2019 06:18:09	Cancel Appointment														
<ul style="list-style-type: none">display appropriate message if there is no appointment.	<div><h3>Scheduled Appointment</h3><div>No Records Found</div></div>																		

Requested Appointments

- Should display the booked appointments of all patients.
- Should display selected patient details, when clicking on view details of particular patient.
- Should delete appointment when clicked on cancel appointment button.

Health Care Service

Welcome

User1

Profile

Register patient

Viewall patients

Requested App

Logout

Appointment id	Patient id	Patient name	Disease	Priority	Tentative date	Requested on	Update
13	400	first name	Flu	Normal	21/02/2019	19/02/2019 04:00:06	<div>View Details</div> <div>Cancel Appointment</div>
14	401	second name	fever, pain	Normal	23/10/2019	21/10/2019 01:51:58	<div>View Details</div> <div>Cancel Appointment</div>
15	401	second name	acid reflex	Normal	22/10/2019	22/10/2019 02:53:01	<div>View Details</div> <div>Cancel Appointment</div>
16	403	fourth name	Adenovirus Infection	Urgent	22/10/2019	22/10/2019 02:53:27	<div>View Details</div> <div>Cancel Appointment</div>

Header

<ul style="list-style-type: none"> The header component should be displayed on top of all pages except login page. Should display the logged user name. Clicking on the particular menu should navigate to the corresponding page. 	<div> <div>Health Care Service Welcome User1 Profile Register patient Viewall patients Requested App Logout</div> </div>
---	---

Services

The application has the following services:

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

Other Instructions

- All the project files are kept inside the src folder.
- ***Do not modify the ids or classes*** for the existing UI elements as they are required by the unit tests.
- The api url is predefined in the service as API_URL. You would hit **API_URL + endpoint** as required.