

# **Patient-Medicine-And-Appointment-System**

## **1. Introduction**

**DoctorsCare is an innovative application designed to provide healthcare services online, allowing patients to access medical aid from the comfort of their homes. This platform caters to the growing need for digital healthcare solutions, offering convenience and efficiency to both patients and doctors. As technology evolves rapidly, DoctorsCare aims to bridge the gap between healthcare services and technology, creating a seamless user experience.**

### **Purpose**

**DoctorsCare aims to eliminate traditional healthcare systems' challenges, such as long waiting lines, misplaced medical records, and time-consuming processes. By digitizing healthcare, DoctorsCare not only saves time and cost for patients but also empowers doctors with comprehensive access to patient records for improved treatment.**

### **Scope**

**DoctorsCare is a future-oriented platform designed for individuals who seek:**

- Online medical consultations.**
- Access to previous medical records.**
- The convenience of scheduling appointments and accessing prescriptions digitally.**
- Reduced costs and time spent on healthcare services.**

## **2. Features of DoctorsCare**

### **2.1 Patient-Centric Features**

#### **1. Online Appointments:**

- Patients can book appointments with doctors at their convenience.**
- Avoids long queues in hospitals.**

#### **2. E-Prescriptions:**

- Doctors can provide digital prescriptions, reducing the risk of losing them.**

- Patients can easily access and share prescriptions with pharmacies or other doctors.

### **3. Access to Medical History:**

- Patients' previous medical records, reports, and prescriptions are securely stored in the database.
- Enables easy retrieval and sharing.

### **4. Cost-Effective Consultations:**

- Reduced travel and waiting time.
- Saves on transportation and other expenses.

### **5. Home-Based Healthcare:**

- Patients can consult doctors from their homes, ensuring convenience and comfort.

## **2.2 Doctor-Centric Features**

### **1. Comprehensive Medical Records:**

- Doctors have access to patients' complete medical histories, aiding in accurate diagnosis and better treatment.

### **2. Time Management:**

- Doctors can manage appointments and consultations more effectively.

### **3. Digital Prescriptions:**

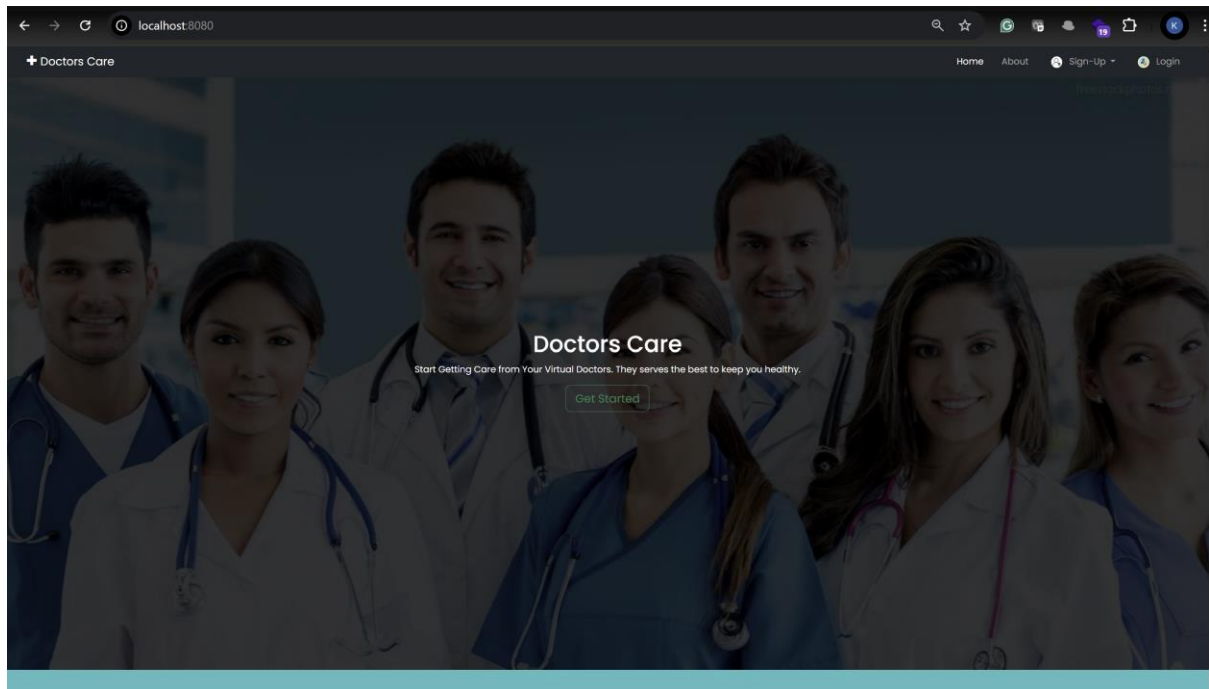
- Reduced paperwork and faster prescription delivery to patients.

### **4. Enhanced Communication:**

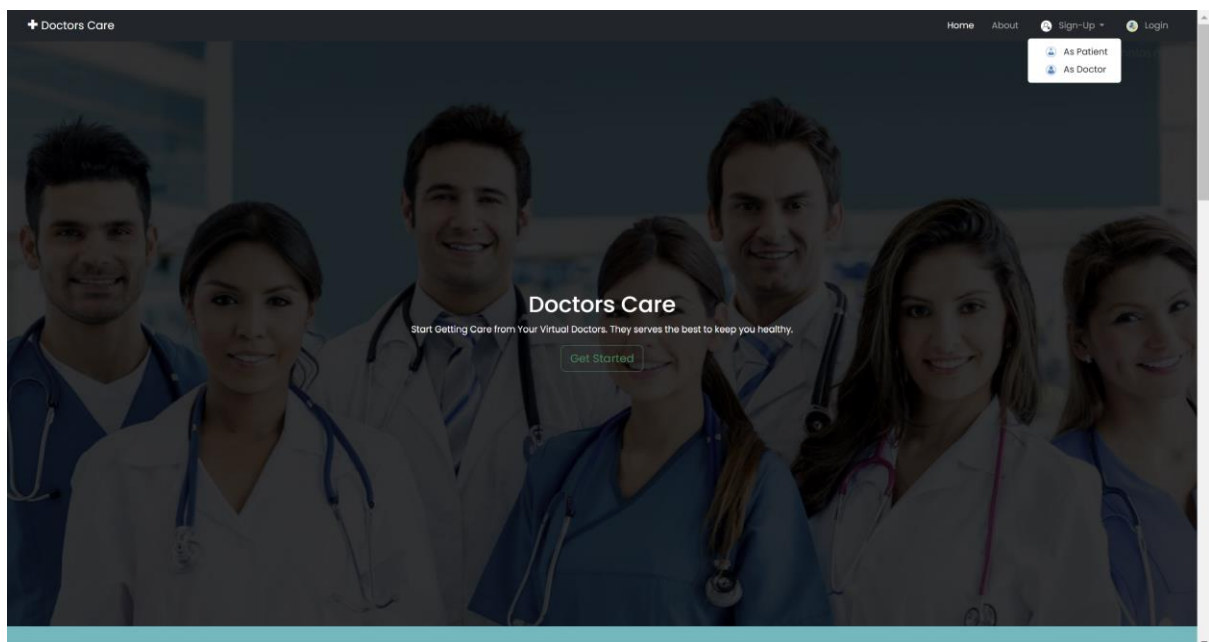
- Direct communication with patients through the platform for follow-ups and clarifications.

## Screenshots:

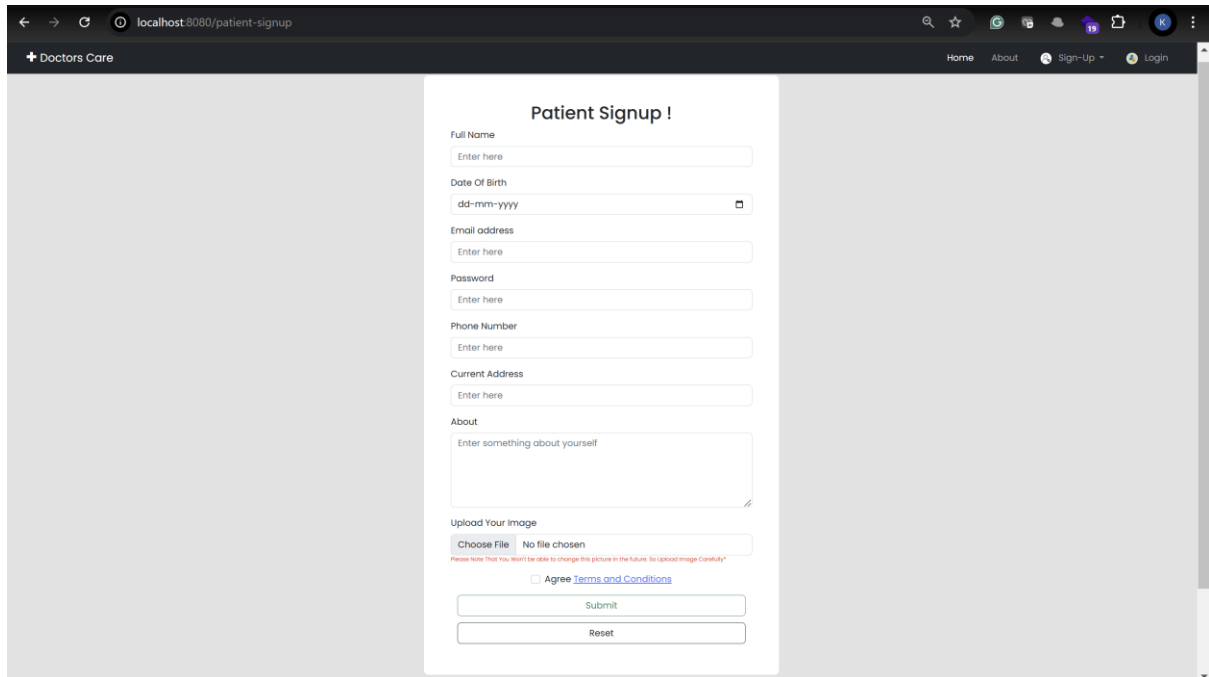
### Home Page:



You can sign up as doctor or patient by clicking signup button



## Sign-up as patient



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/patient-signup'. The page has a dark header with 'Doctors Care' on the left and navigation links 'Home', 'About', 'Sign-Up', and 'Login' on the right. The main content area features a white box titled 'Patient Signup !'. Inside this box, there are several input fields: 'Full Name' (with a placeholder 'Enter here'), 'Date Of Birth' (with a placeholder 'dd-mm-yyyy' and a calendar icon), 'Email address' (with a placeholder 'Enter here'), 'Password' (with a placeholder 'Enter here'), 'Phone Number' (with a placeholder 'Enter here'), and 'Current Address' (with a placeholder 'Enter here'). Below these is an 'About' section with a text area and a placeholder 'Enter something about yourself'. At the bottom of the form is an 'Upload Your Image' section with a 'Choose File' button, a 'No file chosen' status, and a red warning message: 'Please Note That You Won't be able to change this picture in the future. So Upload Image Carefully!'. There is also a checkbox for 'Agree Terms and Conditions' and two buttons at the very bottom: 'Submit' and 'Reset'.

Doctors Care

Home About Sign-Up Login

### Patient Signup !

Full Name  
Enter here

Date Of Birth  
dd-mm-yyyy

Email address  
Enter here

Password  
Enter here

Phone Number  
Enter here

Current Address  
Enter here

About  
Enter something about yourself

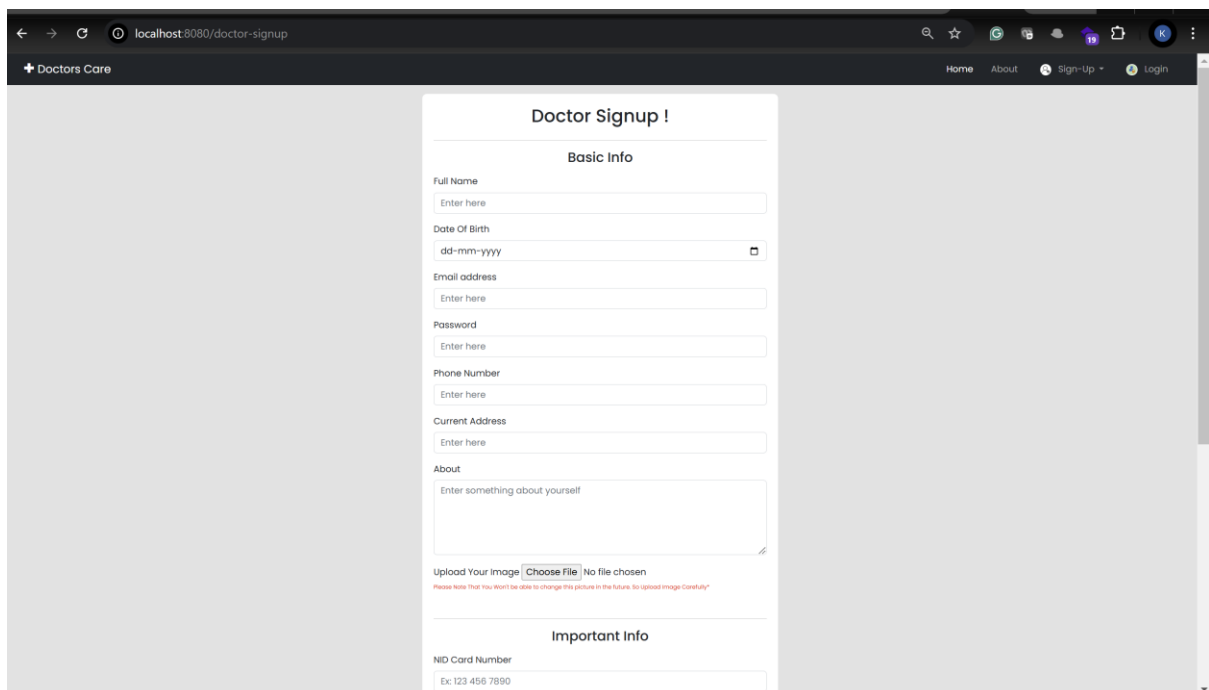
Upload Your Image  
Choose File No file chosen  
Please Note That You Won't be able to change this picture in the future. So Upload Image Carefully!

☐ Agree [Terms and Conditions](#)

Submit

Reset

## As a doctor



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/doctor-signup'. The page has a dark header with 'Doctors Care' on the left and navigation links 'Home', 'About', 'Sign-Up', and 'Login' on the right. The main content area features a white box titled 'Doctor Signup !'. Inside this box, there is a section titled 'Basic Info' which contains the same set of input fields as the patient form: 'Full Name', 'Date Of Birth', 'Email address', 'Password', 'Phone Number', 'Current Address', and 'About'. Below the 'Basic Info' section is an 'Upload Your Image' section with a 'Choose File' button, a 'No file chosen' status, and a red warning message: 'Please Note That You Won't be able to change this picture in the future. So Upload Image Carefully!'. At the bottom of the form is a section titled 'Important Info' which contains a 'NID Card Number' input field with a placeholder 'Ex: 123 456 7890'.

Doctors Care

Home About Sign-Up Login

### Doctor Signup !

#### Basic Info

Full Name  
Enter here

Date Of Birth  
dd-mm-yyyy

Email address  
Enter here

Password  
Enter here

Phone Number  
Enter here

Current Address  
Enter here

About  
Enter something about yourself

Upload Your Image  
Choose File No file chosen  
Please Note That You Won't be able to change this picture in the future. So Upload Image Carefully!

#### Important Info

NID Card Number  
Ex: 123 456 7890

localhost:8080/doctor-signup

Doctors Care

Home About Sign-Up Login

Enter something about yourself

Upload Your Image [Choose File](#) No file chosen  
Please Note That You Won't be able to change the picture in the future. So Upload Image Carefully!

**Important Info**

NID Card Number  
Ex: 123 456 7890

Specialist at  
(select one)

Degrees  
Ex: MBBS, FCPS, Surgeon

Medical College Name  
Ex: Rajshahi Medical College

Appointment Fee tk (Tk)  
Ex: 100

☐ Agree [Terms and Conditions](#)

Submit

Reset

f t G @ in

## Sign-in/ Log-in

localhost:8080/login

Doctors Care

Home About Sign-Up Login

**Please sign in!**

User Email  
User Email

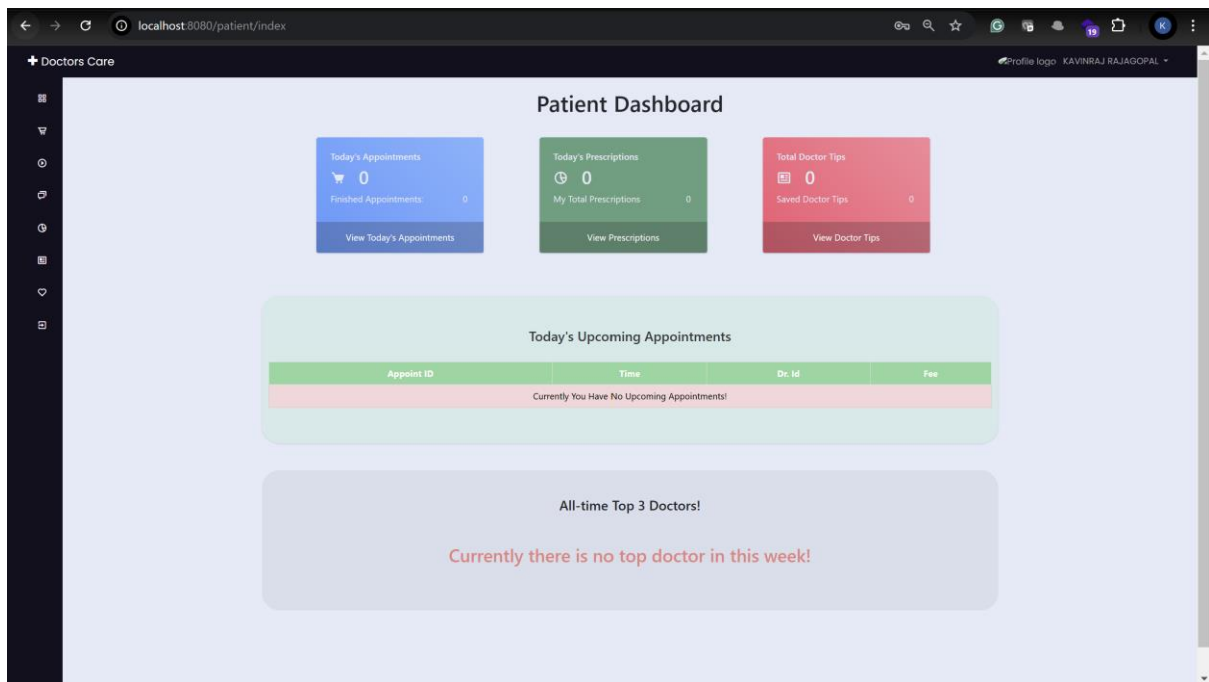
Password  
Password

[Sign In](#)

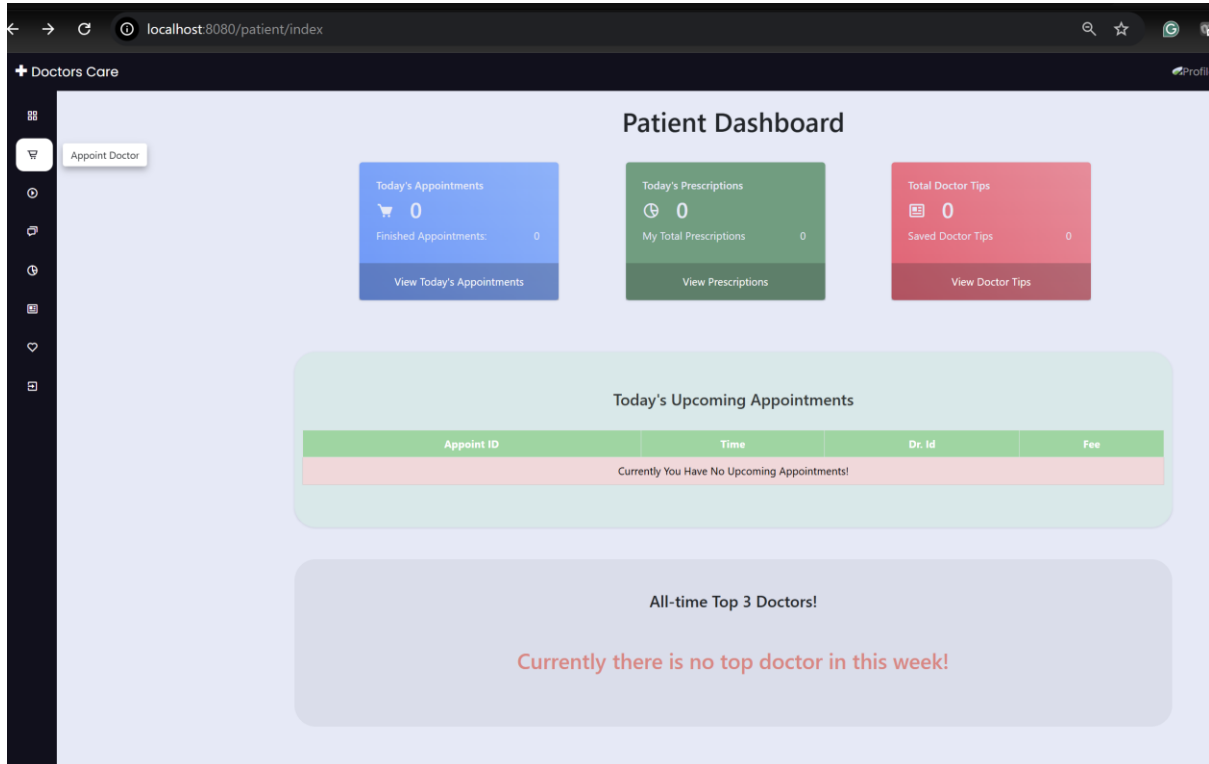
[Forgot Password?](#)

f t G @ in

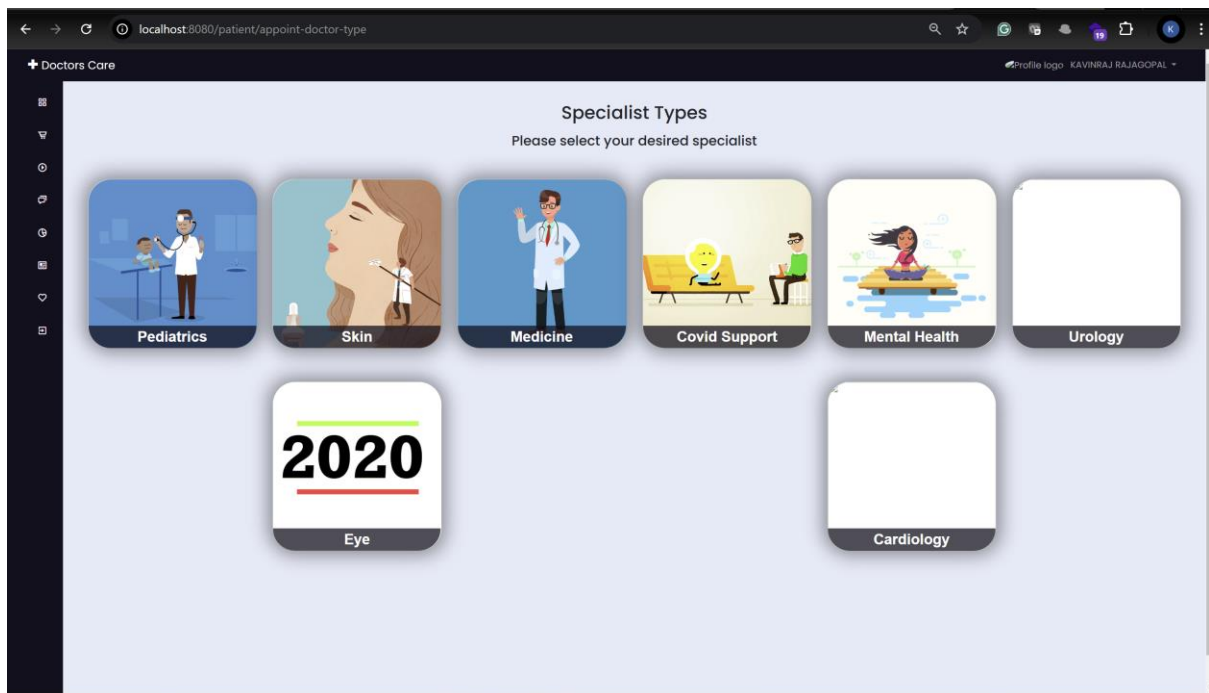
## Patient Dashboard



To book an appointment click the appoint doctor button from left side pannel

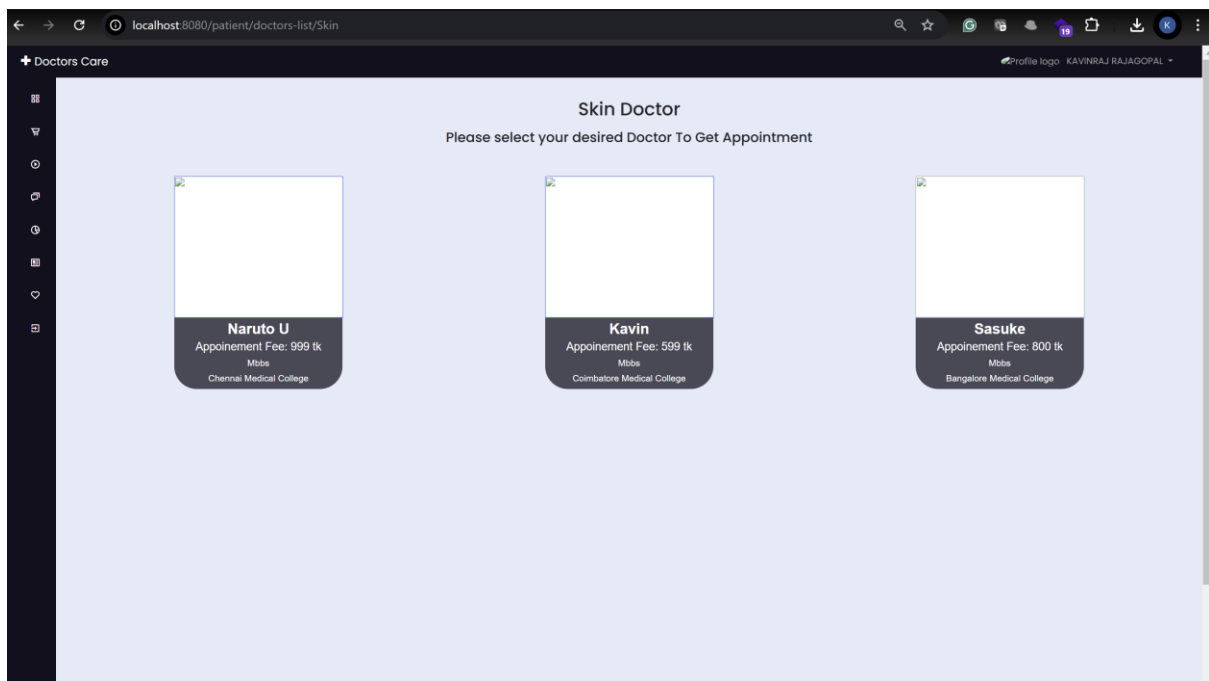


You will be land on the below page

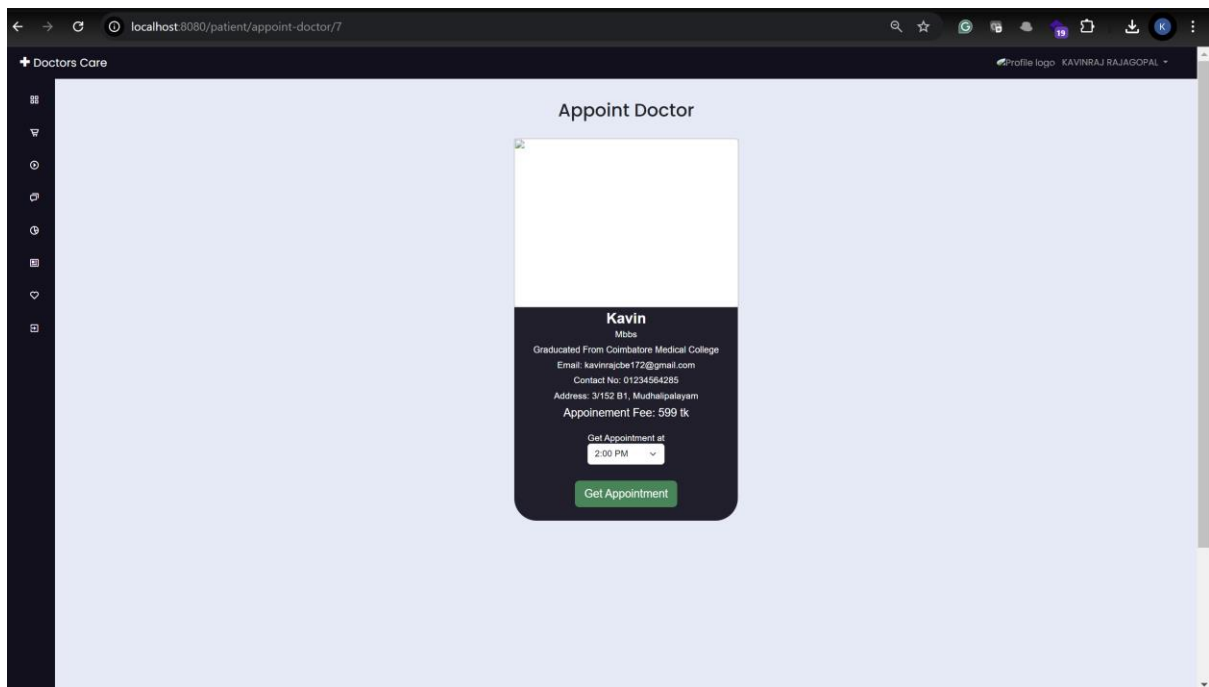


Now it's patients can choose the specialist

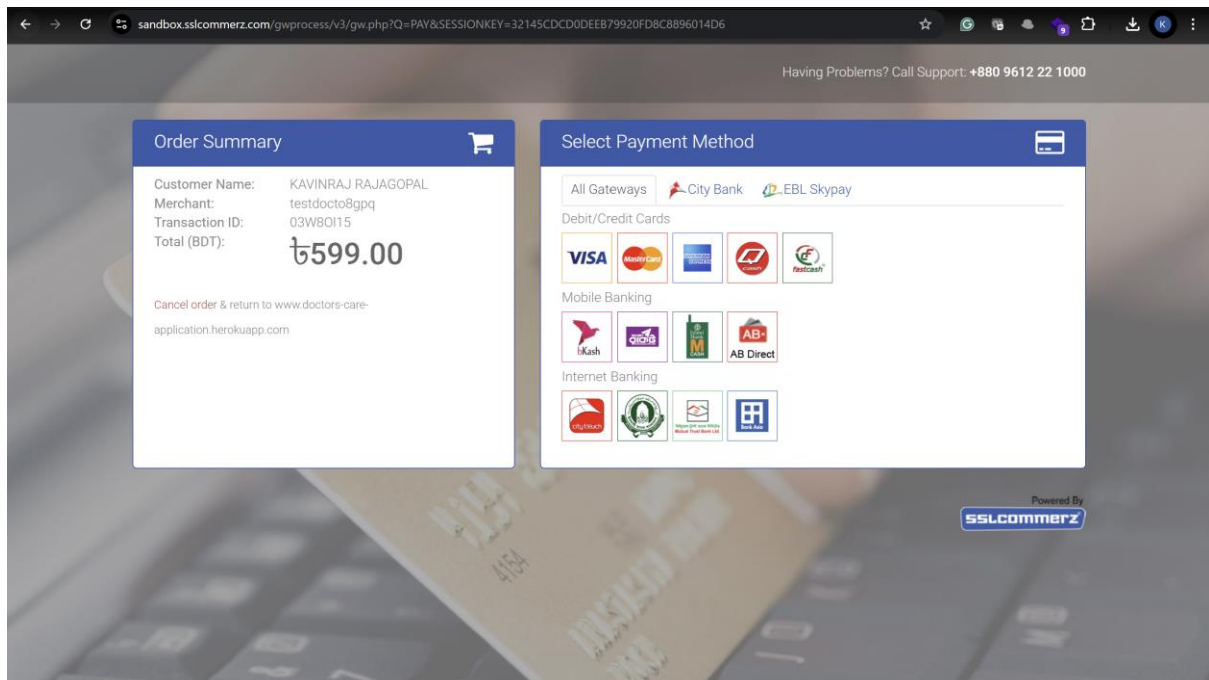
Currently, I have registered 3 skin specialist



I choose Doctor Kavin and booked and appointment for 2 pm

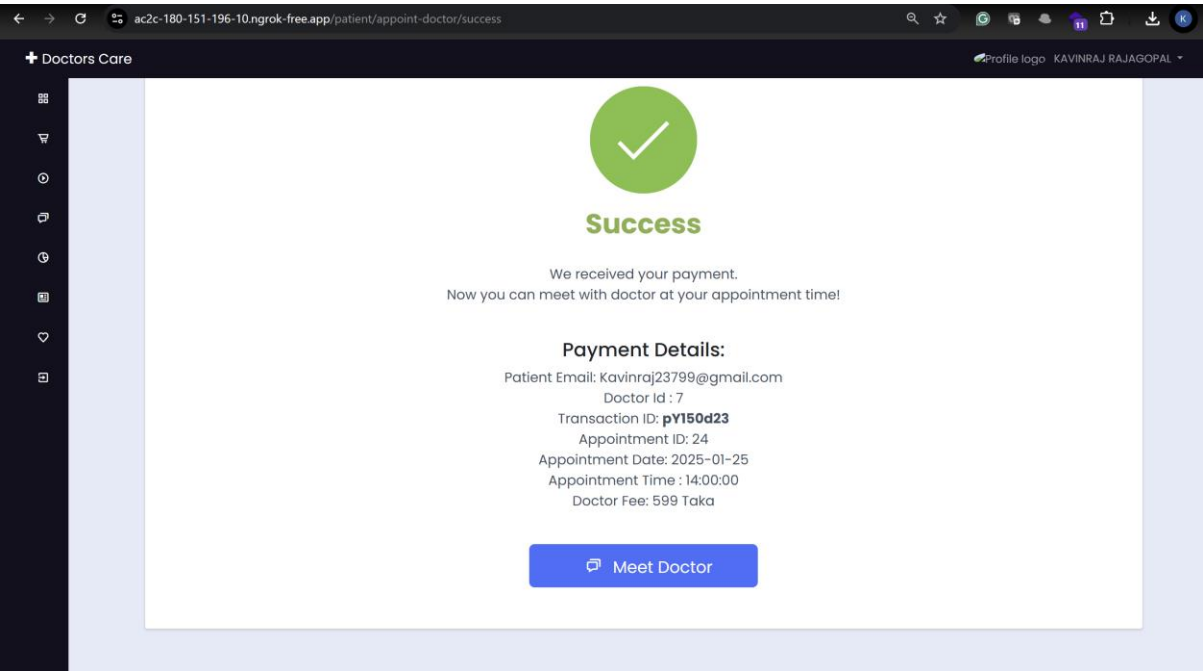


Next step is payment Process





Once the payment is made patient will see below image



The patient can see their appointment by clicking today's appointment button



By clicking Prescription they can see their Prescription

Appoint ID	Date	Time	Dr. ID	Symptoms	Action
29	2025-01-25	15:00:00	11	scars	<a href="#">View</a>
27	2025-01-25	14:00:00	3	open pores	<a href="#">View</a>
24	2025-01-25	14:00:00	7	Dry Skin	<a href="#">View</a>

The patient can view their detailed prescription as below, they can download and print the document

Appointment ID: 24

**Dr. Kavim**  
Mbbbs  
Coimbatore Medical College  
01234564285

**Patient Details:**  
**KAVINRAJ RAJAGOPAL**  
3/152 BL,  
Mudhalpalayam  
DOB: 1999-07-23  
7548842893

**Symptoms**  
Dry Skin

**Medicines**  
**R<sub>x</sub>**  
Vitamin C Glow Tablet

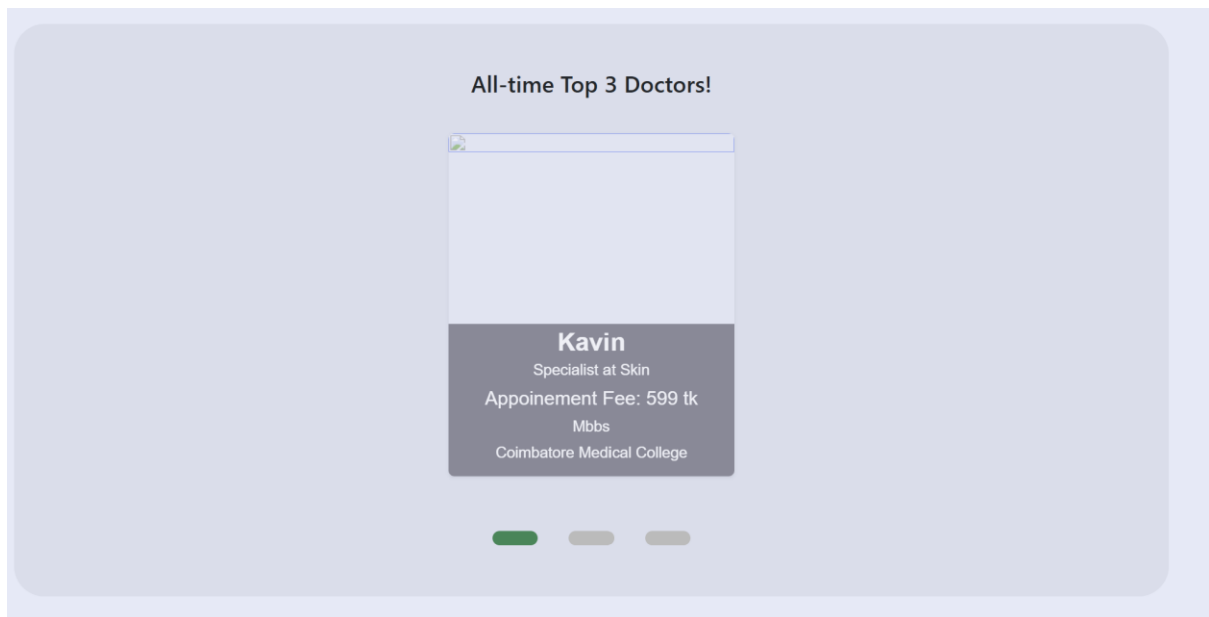
**Tests**  
2

**Advice**  
Proper Sleep is required

Prescribed By  
Kavin  
Date: 2025-01-25

[Print](#)

**Patients can see the top three doctors in the hospital**



### **3. Benefits of DoctorsCare**

#### **3.1 For Patients**

- **Convenience:** No need to physically visit the hospital for minor issues or routine follow-ups.
- **Data Security:** All medical records are securely stored in the cloud, reducing the risk of loss or damage.
- **Accessibility:** 24/7 access to medical records and prescriptions.
- **Time-Saving:** Avoid long waiting periods at hospitals or clinics.
- **Affordable Healthcare:** Saves on travel and other incidental expenses.

#### **3.2 For Doctors**

- **Efficient Workflow:** Streamlines patient management with access to their medical history in one place.
- **Better Treatment Decisions:** Comprehensive medical records ensure informed decisions.

- **Paperless Workflow:** Reduces administrative burden by digitizing prescriptions and reports.

## **4. Challenges Addressed by DoctorsCare**

### **4.1 Traditional Healthcare Issues**

#### **1. Long Waiting Times:**

- Patients often wait in long lines for consultations. DoctorsCare eliminates this by enabling online appointments.

#### **2. Misplaced Medical Records:**

- In the traditional system, losing prescriptions or reports was common. DoctorsCare ensures all records are securely stored and easily accessible.

#### **3. Inconvenience for Patients:**

- Patients, especially those with mobility challenges, often face difficulties visiting hospitals. DoctorsCare provides healthcare from the comfort of their homes.

### **4.2 Modern-Day Expectations**

#### **1. Time Efficiency:**

- People value their time more than ever. DoctorsCare respects this by minimizing unnecessary delays.

#### **2. Cost Reduction:**

- Digital consultations are cost-effective, saving on transportation and related expenses.

## **5. System Design and Architecture**

### **5.1 High-Level Architecture**

**DoctorsCare consists of three primary modules:**

#### **1. User Module:**

- Handles patient and doctor registration, login, and profile management.

## **2. Appointment Module:**

- **Manages online appointment scheduling, cancellation, and reminders.**

## **3. Medical Records Module:**

- **Stores and retrieves medical reports, prescriptions, and consultation notes.**

## **5.2 Database Design**

- **Patient Table:**
  - **Stores patient details such as name, contact info, and medical history.**
- **Doctor Table:**
  - **Stores doctor details including specialization, experience, and availability.**
- **Appointment Table:**
  - **Tracks appointment details such as date, time, and status.**
- **Medical Records Table:**
  - **Securely stores prescriptions and test reports.**

## **5.3 Security**

**DoctorsCare ensures data privacy and security through:**

- **Role-based access control (RBAC) for patients and doctors.**
- **Encryption of sensitive data.**
- **Compliance with healthcare data regulations like HIPAA (if applicable).**

## **6. Technologies Used**

### **6.1 Frontend**

- **HTML5, CSS3, JavaScript: For user interface design.**
- **ReactJS/AngularJS: For building dynamic and interactive web pages.**

### **6.2 Backend**

- **Spring Boot: For handling business logic and API development.**

- **Java:** Primary programming language for backend development.

### **6.3 Database**

- **MySQL/PostgreSQL:** For secure and efficient data storage.

### **6.4 Deployment**

- **Cloud Platform:** AWS, Azure, or Google Cloud for hosting the application.
- **Docker:** For containerization and seamless deployment.

## **7. Implementation Plan**

### **7.1 Milestones**

- 1. Phase 1: Requirement Analysis and Design**
  - Identify core features and finalize system architecture.
- 2. Phase 2: Development**
  - Develop frontend, backend, and integrate the database.
- 3. Phase 3: Testing**
  - Perform unit, integration, and user acceptance testing (UAT).
- 4. Phase 4: Deployment**
  - Deploy the application to a cloud platform for public use.
- 5. Phase 5: Maintenance**
  - Regular updates and bug fixes.

### **7.2 Timeline**

- **Phase 1: 2 weeks**
- **Phase 2: 6 weeks**
- **Phase 3: 2 weeks**
- **Phase 4: 1 week**
- **Phase 5: Ongoing**

# SWAGGER API DETAILS

## 1 HOME CONTROLLER

### PATIENT SIGNUP:

home-controller

POST /process-patient-signup

Try it out

Name	Description
user <small>required</small> object (query)	<pre>{   "id": 0,   "name": "string",   "email": "string",   "password": "string",   "about": "string",   "role": "string",   "enabled": true,   "imageUrl": "string",   "phone": "string",   "address": "string",   "posts": [     {       "id": 0,       "postId": "2025-01-25T07:57:59.783Z",       "postTime": "2025-01-25T07:57:59.783Z",       "coverPhoto": "string",       "likesList": [         {           "id": 0         }       ]     }   ] }</pre>

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

Payload:

```
{
  "id": 0,
  "name": "string",
  "email": "string",
  "password": "string",
  "about": "string",
  "role": "string",
  "enabled": true,
  "imageUrl": "string",
  "phone": "string",
  "address": "string",
  "posts": [
```



```
{
  "id": 0,
  "postDate": "2025-01-25T07:57:59.783Z",
  "postTime": "2025-01-25T07:57:59.783Z",
  "coverPhoto": "string",
  "likesList": [
    {
      "id": 0,
      "likerId": "string",
      "post": "string"
    }
  ],
  "commentsList": [
    {
      "id": 0,
      "commenterId": "string",
      "commenterName": "string",
      "commenterImage": "string",
      "comment": "string",
      "posts": "string"
    }
  ],
  "savedPostsList": [
    {
      "id": 0,
      "saverId": "string",
      "posts": "string"
    }
  ],
  "user": "string",
  "postContent": "string"
```

```
}  
],  
"doctorsAdditionalInfo": {  
  "id": 0,  
  "nid": "string",  
  "doctortype": "string",  
  "degrees": "string",  
  "medicalcollege": "string",  
  "appointmentfee": "string",  
  "user": "string"  
},  
"doctorsSchedule": {  
  "id": 0,  
  "is_10_00": true,  
  "is_10_30": true,  
  "is_11_00": true,  
  "is_11_30": true,  
  "is_12_00": true,  
  "is_2_00": true,  
  "is_2_30": true,  
  "is_3_00": true,  
  "is_3_30": true,  
  "user": "string"  
},  
"accountActiveToken": {  
  "id": 0,  
  "token": "string",  
  "tokenGenerationDate": "2025-01-25T07:57:59.783Z",  
  "user": "string",  
  "emailsVerified": true  
},
```

```
"dob": "string"
}
```

## DOCTOR SIGNUP

POST /process-doctor-signup

Parameters

Try it out

Name	Description
doctorSignup * required	
object (query)	<pre>{   "name": "string",   "email": "kiitwlnpq54ya6649jxb~R==W &amp;3n^D0@aE3\$O@cqlksifiyrcsdaoazdkkwaxeqvgvjxazinwvyrldlfbxaxd.liw",   "password": "string",   "about": "string",   "role": "string",   "enabled": true,   "imageUrl": "string",   "phone": "string",   "address": "string",   "nid": "string",   "doctortype": "string",   "degrees": "string",   "medicalcollege": "string",   "appointmentfee": "string",   "dob": "string" }</pre>

Responses

Code	Description	Links
200	OK	No links

Media type: \*/\* (v)

Controls Accept header.

Example Value: string

Payload:

```
{
  "name": "string",
  "email":
  "kiitwlnpq54ya6649jxb~R==W|&3n^D0@aE3$O@cqlksifiyrcsdaoazdkkwaxeqvgvjxazinwvyrldlfbxaxd.liw",
  "password": "string",
  "about": "string",
  "role": "string",
  "enabled": true,
  "imageUrl": "string",
  "phone": "string",
  "address": "string",
  "nid": "string",
  "doctortype": "string",
  "degrees": "string",
  "medicalcollege": "string",
}
```

```
"appointmentfee": "string",

"dob": "string"
}
```

## CHANGE PASSWORD

POST

/change-password

Parameters

Try it out

Name	Description
<b>newPassword</b> * required string (query)	<input type="text" value="newPassword"/>
<b>confirmPassword</b> * required string (query)	<input type="text" value="confirmPassword"/>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/"/></div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## GET OTP

GET

/verify-otp

Parameters

Try it out

Name	Description
<b>OTP</b> * required string (query)	<input type="text" value="OTP"/>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/"/></div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

VERIFY OTP

GET

/verify-account

Try it out

Parameters

Name	Description
token * required string (query)	token

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

GET

/test-nextId

Try it out

Parameters

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

TERMS AND CONDITION

GET

/terms-and-conditions

Try it out

Parameters

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## SEND OTP

GET

/send-otp

^

Parameters

Try it out

Name	Description
email <span>required</span>	
string (query)	email

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

## PATIENT SIGNUP

GET

/patient-signup

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

## LOGIN

GET

/login

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

## HANDLE PAYMENT

GET

/handle-payment

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{   "applicationContext": {     "parent": "string",     "id": "string",     "displayName": "string",     "applicationName": "string",     "startupDate": 0,     "autowireCapableBeanFactory": {},     "environment": {       "defaultProfiles": [         "string"       ],       "activeProfiles": [         "string"       ]     },     "beanDefinitionCount": 0,     "beanDefinitionNames": [       "string"     ],     "parentBeanFactory": {},     "classloader": {       "name": "string",       "registeredAsParallelCapable": true,       "parent": {         "name": "string",         "registeredAsParallelCapable": true, </pre>	

No links

EG:

```
{
  "applicationContext": {
    "parent": "string",
    "id": "string",
    "displayName": "string",
    "applicationName": "string",
    "startupDate": 0,
    "autowireCapableBeanFactory": {},
    "environment": {
      "defaultProfiles": [
        "string"
      ],
      "activeProfiles": [
        "string"
      ]
    }
  }
}
```

```
},
"beanDefinitionCount": 0,
"beanDefinitionNames": [
  "string"
],
"parentBeanFactory": {},
"classLoader": {
  "name": "string",
  "registeredAsParallelCapable": true,
  "parent": {
    "name": "string",
    "registeredAsParallelCapable": true,
    "unnamedModule": {
      "name": "string",
      "classLoader": {
        "name": "string",
        "registeredAsParallelCapable": true,
        "definedPackages": [
          {
            "name": "string",
            "annotations": [
              {}
            ],
            "declaredAnnotations": [
              {}
            ],
            "sealed": true,
            "specificationTitle": "string",
            "specificationVersion": "string",
            "specificationVendor": "string",
            "implementationTitle": "string",
```



```
    "implementationVersion": "string",
    "implementationVendor": "string"
  }
]
},
"descriptor": {
  "open": true,
  "automatic": true
},
"named": true,
"annotations": [
  {}
],
"declaredAnnotations": [
  {}
],
"packages": [
  "string"
],
"nativeAccessEnabled": true,
"layer": {}
},
"definedPackages": [
  {
    "name": "string",
    "annotations": [
      {}
    ],
    "declaredAnnotations": [
      {}
    ],
  },
]
```

```
"sealed": true,
"specificationTitle": "string",
"specificationVersion": "string",
"specificationVendor": "string",
"implementationTitle": "string",
"implementationVersion": "string",
"implementationVendor": "string"
}
]
},
"unnamedModule": {
"name": "string",
"classLoader": {
"name": "string",
"registeredAsParallelCapable": true,
"definedPackages": [
{
"name": "string",
"annotations": [
{}
],
"declaredAnnotations": [
{}
],
"sealed": true,
"specificationTitle": "string",
"specificationVersion": "string",
"specificationVendor": "string",
"implementationTitle": "string",
"implementationVersion": "string",
"implementationVendor": "string"
```

```
    }  
  ]  
},  
"descriptor": {  
  "open": true,  
  "automatic": true  
},  
"named": true,  
"annotations": [  
  {}  
],  
"declaredAnnotations": [  
  {}  
],  
"packages": [  
  "string"  
],  
"nativeAccessEnabled": true,  
"layer": {}  
},  
"definedPackages": [  
  {  
    "name": "string",  
    "annotations": [  
      {}  
    ],  
    "declaredAnnotations": [  
      {}  
    ],  
    "sealed": true,  
    "specificationTitle": "string",
```

```
    "specificationVersion": "string",
    "specificationVendor": "string",
    "implementationTitle": "string",
    "implementationVersion": "string",
    "implementationVendor": "string"
  }
]
},
"contentType": "string",
"requestContextAttribute": "string",
"staticAttributes": {
  "additionalProp1": {},
  "additionalProp2": {},
  "additionalProp3": {}
},
"exposePathVariables": true,
"beanName": "string",
"url": "string",
"hosts": [
  "string"
],
"redirectView": true,
"propagateQueryProperties": true,
"attributesMap": {
  "additionalProp1": {},
  "additionalProp2": {},
  "additionalProp3": {}
}
}
```

## FORGOR PASSWORD

GET

/forgot-password

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div>string</div>	<div>No links</div>

## DOCTOR SIGNUP

GET

/doctor-signup

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div>string</div>	<div>No links</div>

## 2. APPOINT DOCTOR VALIDATOR

appoint-doctor-validator

POST

/pay-success-validator

Parameters

Try it out

Name	Description
<div><div>requestMap</div><div>object</div><div>(query)</div></div>	<div><div><div>required</div></div><div><pre>{   "additionalProp1": "string",   "additionalProp2": "string",   "additionalProp3": "string" }</pre></div></div>

Responses

Code	Description	Links
200	<div><div>OK</div><div>Media type</div><div>*/*</div><div>Controls Accept header.</div><div>Example Value   Schema</div><div>string</div></div>	No links

## 3. PATIENT MAIN CONTROLLER

### VIEW SINGLE PRESCRIPTION

patient-main-controller

POST

/patient/view-single-prescription

Parameters

Try it out

Name	Description
<div><div>appointmentID</div><div>string</div><div>(query)</div></div>	<div><div><div>required</div></div><div>appointmentID</div></div>

Responses

Code	Description	Links
200	<div><div>OK</div><div>Media type</div><div>*/*</div><div>Controls Accept header.</div><div>Example Value   Schema</div><div>string</div></div>	No links

VIEW PRESCRIPTION BASED ON PAGE NUMBER

GET

/patient/view-prescriptions/{page}

Parameters

Try it out

Name	Description
page * required integer(\$int32) (path)	<div>page</div>

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>application/*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

UPLOAD IMAGE

GET

/patient/uploadedImages/{filename}

Parameters

Try it out

Name	Description
filename * required string (path)	<div>filename</div>

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>application/octet-stream</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>[ "string" ]</div>	No links

TODAY'S APPOINTMENT

GET

/patient/todays-appointment

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## PATIENT PROFILE

GET

/patient/profile

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## MEET DOCTOR

GET

/patient/meet-doctor

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links



GET

/patient/index

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*<div></div><div>Controls Accept header.</div><div>Example Value   Schema</div><div>string</div></div>	No links

GET

/patient/appoint-doctor-type

^

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK <div>Media type */*<div></div><div>Controls Accept header.</div><div>Example Value   Schema</div><div>string</div></div>	No links

patient-doctors-t-ips

^

POST

/patient/process-unsavepost

^

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value | Schema

{  
 "postId": "string",  
 "unsaverId": "string"  
}

Responses

Code	Description	Links
200	OK <div>Media type */*<div></div><div>Controls Accept header.</div><div>Example Value   Schema</div><div>{}</div></div>	No links

POST

/patient/process-savepost

^

Parameters

Try it out

No parameters

Request body

required

application/json

▼

Example Value

Schema

```
{
  "saverId": "string",
  "postId": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
{}
```

POST

/patient/process-like

^

Parameters

Try it out

No parameters

Request body

required

application/json

▼

Example Value

Schema

```
{
  "postId": "string",
  "likerId": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
{}
```

POST

/patient/process-comment

Try it out

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "postId": "string",
  "commenterId": "string",
  "commenterName": "string",
  "commenterImage": "string",
  "comment": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

```
{}
```

GET

/patient/saved-tips-posts/{page}

Try it out

Parameters

Name	Description
page * <small>required</small> integer(\$int32) (path)	page

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

```
string
```

GET

/patient/post-homepage/{page}

Try it out

Parameters

Name	Description
page * <small>required</small> integer(\$int32) (path)	page

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

```
string
```

4. PATIENT APPOINT DOCTOR CONTROLLER

POST

/patient/appoint-doctor/success

Parameters

Try it out

Name

Description

requestMap

object

(query)

required

{

additionalProp1: string,

additionalProp2: string,

additionalProp3: string

}

Responses

Code

Description

Links

200

OK

No links

Media type

\*/\*

Controls Accept header.

Example Value

Schema

string

POST

/patient/appoint-doctor/attempt-payment

Parameters

Try it out

Name

Description

appointDoctorDTO

object

(query)

required

{

appointmentTime: "2025-01-25T08:21:30.066Z",

doctorID: string,

doctorFee: string

}

Responses

Code

Description

Links

200

OK

No links

Media type

\*/\*

Controls Accept header.

Example Value

Schema

{

applicationContext: {

parent: string,

id: string,

displayName: string,

applicationName: string,

startDate: 0,

autowireCapableBeanFactory: {},

environment: {

defaultProfiles: [

string

}

}

}

Response:

```
{
  "applicationContext": {
    "parent": "string",
    "id": "string",
    "displayName": "string",
    "applicationName": "string",
    "startupDate": 0,
    "autowireCapableBeanFactory": {},
    "environment": {
      "defaultProfiles": [
        "string"
      ],
      "activeProfiles": [
        "string"
      ]
    },
    "beanDefinitionCount": 0,
    "beanDefinitionNames": [
      "string"
    ],
    "parentBeanFactory": {},
    "classLoader": {
      "name": "string",
      "registeredAsParallelCapable": true,
      "parent": {
        "name": "string",
        "registeredAsParallelCapable": true,
        "unnamedModule": {
          "name": "string",
          "classLoader": {
            "name": "string",
```

```
"registeredAsParallelCapable": true,
"definedPackages": [
  {
    "name": "string",
    "annotations": [
      {}
    ],
    "declaredAnnotations": [
      {}
    ],
    "sealed": true,
    "specificationTitle": "string",
    "specificationVersion": "string",
    "specificationVendor": "string",
    "implementationTitle": "string",
    "implementationVersion": "string",
    "implementationVendor": "string"
  }
],
"descriptor": {
  "open": true,
  "automatic": true
},
"named": true,
"annotations": [
  {}
],
"declaredAnnotations": [
  {}
],
```

```
"packages": [  
  "string"  
],  
"nativeAccessEnabled": true,  
"layer": {}  
},  
"definedPackages": [  
  {  
    "name": "string",  
    "annotations": [  
      {}  
    ],  
    "declaredAnnotations": [  
      {}  
    ],  
    "sealed": true,  
    "specificationTitle": "string",  
    "specificationVersion": "string",  
    "specificationVendor": "string",  
    "implementationTitle": "string",  
    "implementationVersion": "string",  
    "implementationVendor": "string"  
  }  
]  
},  
"unnamedModule": {  
  "name": "string",  
  "classLoader": {  
    "name": "string",  
    "registeredAsParallelCapable": true,  
    "definedPackages": [  

```

```
{
  "name": "string",
  "annotations": [
    {}
  ],
  "declaredAnnotations": [
    {}
  ],
  "sealed": true,
  "specificationTitle": "string",
  "specificationVersion": "string",
  "specificationVendor": "string",
  "implementationTitle": "string",
  "implementationVersion": "string",
  "implementationVendor": "string"
}
],
},
"descriptor": {
  "open": true,
  "automatic": true
},
"named": true,
"annotations": [
  {}
],
"declaredAnnotations": [
  {}
],
"packages": [
  "string"
```



```
],  
  "nativeAccessEnabled": true,  
  "layer": {}  
},  
"definedPackages": [  
  {  
    "name": "string",  
    "annotations": [  
      {}  
    ],  
    "declaredAnnotations": [  
      {}  
    ],  
    "sealed": true,  
    "specificationTitle": "string",  
    "specificationVersion": "string",  
    "specificationVendor": "string",  
    "implementationTitle": "string",  
    "implementationVersion": "string",  
    "implementationVendor": "string"  
  }  
]  
}  
},  
"contentType": "string",  
"requestContextAttribute": "string",  
"staticAttributes": {  
  "additionalProp1": {},  
  "additionalProp2": {},  
  "additionalProp3": {}  
},
```

```

"exposePathVariables": true,

"beanName": "string",

"url": "string",

"hosts": [

  "string"

],

"redirectView": true,

"propagateQueryProperties": true,

"attributesMap": {

  "additionalProp1": {},

  "additionalProp2": {},

  "additionalProp3": {}

}

}

```

GET
/patient/doctors-list/{doctortype}
^

Parameters
Try it out

Name	Description
<b>doctortype</b> <small>required</small> string (path)	<input type="text" value="doctortype"/>

Responses

Code	Description	Links
200	OK Media type <input type="text" value="*/"/> Controls Accept header. Example Value   Schema <div>string</div>	No links

GET

/patient/appoint-doctor/{doctorID}

⌵

Parameters

Try it out

Name	Description
<b>doctorID</b> * required Integer (int32) (path)	<input type="text" value="doctorID"/>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/*"/> Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## 5 DOCTOR PRESCRIPTION CONTROLLER

doctor-prescription-controller

⌵

POST

/doctor/save-prescription

⌵

Parameters

Try it out

Name	Description
<b>doctorGivePrescriptionDTO</b> * required object (query)	<div><pre>{   "appointmentID": "string",   "symptoms": "string",   "tests": "string",   "advice": "string",   "medicines": "string" }</pre></div>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/*"/> Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

POST

/doctor/give-prescription

Parameters

Try it out

Name	Description
appointmentID * required	appointmentID
string (query)	

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

## 6. DOCTOR POST HOMEPAGE CONTROLLER

doctor-post-homepage-controller

POST

/doctor/process-unsavepost

Parameters

Try it out

No parameters

Request body \* required

application/json

Example Value | Schema

```
{  "postId": "string",  "unsaverId": "string"}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

{}

POST

/doctor/process-savepost

^

Parameters

Try it out

No parameters

Request body

required

application/json

▼

Example Value

Schema

```
{
  "serverId": "string",
  "postId": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
{}
```

POST

/doctor/process-save-editedpost

^

Parameters

Try it out

Name	Description
<b>postId</b> * required string (query)	<input type="text" value="postId"/>
<b>postContent</b> * required string (query)	<input type="text" value="postContent"/>

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
string
```

POST

/doctor/process-like

⌵

Parameters

Try it out

No parameters

Request body

required

application/json

⌵

Example Value

Schema

```
{
  "postId": "string",
  "likerId": "string"
}
```

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div><div><pre>{}</pre></div></div>	

 No links |

POST

/doctor/process-deletepost

⌵

Parameters

Try it out

No parameters

Request body

required

application/json

⌵

Example Value

Schema

```
{
  "postId": "string"
}
```

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div><div><pre>{}</pre></div></div>	

 No links |

POST

/doctor/process-comment

^

Parameters

Try it out

No parameters

Request body

required

application/json

▼

Example Value

Schema

```
{
  "postId": "string",
  "commenterId": "string",
  "commenterName": "string",
  "commenterImage": "string",
  "comment": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
{}
```

POST

/doctor/edit-post

^

Parameters

Try it out

Name	Description
<b>postId</b> * required string (query)	<input type="text" value="postId"/>

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

▼

Controls Accept header.

Example Value

Schema

```
string
```

POST

/doctor/do-post-homepage

Try it out

Parameters

Name	Description
<b>posts</b> <small>* required</small> object (query)	<pre>{   "id": 0,   "postDate": "2025-01-25T08:49:06.270Z",   "postTime": "2025-01-25T08:49:06.270Z",   "coverPhoto": "string",   "likesList": [     {       "id": 0,       "likerId": "string",       "post": "string"     }   ],   "commentsList": [     {       "id": 0,       "commenterId": "string",       "commenterName": "string",       "commenterImage": "string",       "comment": "string",       "timestamp": "string"     }   ] }</pre>
<b>doctorID</b> <small>* required</small> string (query)	<input type="text" value="doctorID"/>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/"/> Controls Accept header. Example Value   Schema <div>string</div></div>	No links

GET

/doctor/saved-tips-posts/{page}

Try it out

Parameters

Name	Description
<b>page</b> <small>* required</small> integer(\$int32) (path)	<input type="text" value="page"/>

Responses

Code	Description	Links
200	OK <div>Media type <input type="text" value="*/"/> Controls Accept header. Example Value   Schema <div>string</div></div>	No links



GET

/doctor/post-homepage/{page}

⌵

Parameters

Try it out

Name	Description
page <span>required</span>	
integer(\$int32)	page
(path)	

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

admin-patient-doctor-list-edit-delete

⌵

POST

/admin/process-patient-edit

⌵

Parameters

Try it out

Name	Description
adminPatientEditDTO <span>required</span>	
object	
(query)	<pre>{   "id": "string",   "enabled": true,   "name": "string",   "page": "string",   "email": "string",   "phone": "string",   "address": "string",   "about": "string",   "dob": "string" }</pre>

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

string

POST

/admin/process-patient-delete

Try it out

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "userId": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

```
()
```

POST

/admin/process-doctor-edit

Try it out

Parameters

Name	Description
adminDoctorEditDTO <small>required</small>	

object  
(query)

```
{
  "id": "string",
  "enabled": true,
  "name": "string",
  "page": "string",
  "email": "string",
  "phone": "string",
  "address": "string",
  "about": "string",
  "nid": "string",
  "doctortype": "string",
  "degrees": "string",
  "medicalcollege": "string",
  "appointmentfee": "string",
  "dob": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value | Schema

```
string
```

POST

/admin/process-doctor-delete

⌵

Parameters

Try it out

No parameters

Request body

required

application/json

⌵

Example Value

Schema

```
{
  "userId": "string"
}
```

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ }</pre></div>	

 No links |

POST

/admin/edit-patient-details

⌵

Parameters

Try it out

Name	Description
<div>editUserDTO</div> <div>object</div> <div>(query)</div>	<div><pre>{   "pageNo": "string",   "userId": "string" }</pre></div>

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>string</pre></div>	<div>No links</div>

POST /admin/edit-doctor-details

Parameters

Try it out

Name	Description
<b>editUserDTO</b> * required object (query)	<pre>{  "pageNo": "string",  "userId": "string"}  </pre>

Responses

Code	Description	Links
200	OK <div>Media type */* Controls Accept header. Example Value   Schema string</div>	No links

## 7. ADMIN MAIN CONTROLLER

admin-main-controller

POST /admin/process-email-send

Parameters

Try it out

Name	Description
<b>sendMailTo</b> * required string (query)	<input type="text" value="sendMailTo"/>
<b>emailSubject</b> * required string (query)	<input type="text" value="emailSubject"/>
<b>emailBody</b> * required string (query)	<input type="text" value="emailBody"/>

Responses

Code	Description	Links
200	OK <div>Media type */* Controls Accept header. Example Value   Schema string</div>	No links

GET /admin/index

Parameters

Try It out

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## 8. DOCTOR MAIN CONTROLLER

doctor-main-controller

GET /doctor/view-prescriptions/{page}

Parameters

Try It out

Name	Description
page <small>* required</small>	page

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

GET /doctor/index

Parameters

Try It out

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <div>string</div>	No links

## 9. ADMIN EARNING CONTROLLER

admin-earnings-controller

GET /admin/earnings

Try it out

Parameters

No parameters

Responses

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>*/*</div> <div>Controls Accept header.</div> <div>Example ValueSchema</div> <div>string</div>	No links

### Schema Details:

Schemas

AccountActiveToken >

Comments >

DoctorsAdditionalInfo >

DoctorsSchedule >

Likes >

Posts >

SavedPosts >

User >

DoctorSignup >

UnsavePostDTO >

SavedPostsDTO >

LikesDTO >

[EditUserDTO](#) >

```
AccountActiveToken √ {
  id
  token
  tokenGenerationDate
  user
}

integer($int64)
string
string($date-time)

User √ {
  id
  name
  email*
  password*

  about
  role
  enabled
  imageURL
  phone
  address
  posts

  integer($int32)
  string
  string
  string
  maxLength: 250
  minLength: 5
  string
  string
  boolean
  string
  string
  string

  √ {Posts √ {
    id
    postDate
    postTime
    coverPhoto
    likesList
    commentsList
    savedPostsList
    user

    √ {Posts √ {
      id
      postDate
      postTime
      coverPhoto
      likesList
      commentsList
      savedPostsList
      user

      User √ {
        id
        name
        email*
        password*
        about
        role
        enabled
        imageURL
        phone
        address
        posts
        doctorsAdditionalInfo
        doctorsSchedule
        accountActiveToken
        dob

        DoctorsAdditionalInfo > {...}
        DoctorsSchedule > {...}
        AccountActiveToken > {...}
        dob > {...}

        postContent > {...}
      }
    }
  }
  doctorsAdditionalInfo DoctorsAdditionalInfo > {...}
  doctorsSchedule DoctorsSchedule > {...}
  accountActiveToken AccountActiveToken > {...}
  dob > {...}
}

emailIsVerified
boolean
```

```

Comments ▾ {
  id > [...]
  commenterId > [...]
  commenterName > [...]
  commenterImage > [...]
  comment > [...]
  posts
    Posts ▾ {
      id > [...]
      postDate > [...]
      postTime > [...]
      coverPhoto > [...]
      likesList > [...]
      commentsList > [...]
      savedPostsList > [...]
      user
        User ▾ {
          id > [...]
          name > [...]
          email* > [...]
          password* > [...]
          about > [...]
          role > [...]
          enabled > [...]
          imageURL > [...]
          phone > [...]
          address > [...]
          posts > [...]
          doctorsAdditionalInfo > {...}
          doctorsSchedule > {...}
          accountActiveToken > {...}
          dob > [...]
        }
      postContent > [...]
    }
  }
}

```



```

DoctorsAdditionalInfo < {
  id > [...]
  nid > [...]
  doctortype > [...]
  degrees > [...]
  medicalcollege > [...]
  appointmentfee > [...]
  user
    User < {
      id > [...]
      name > [...]
      email* > [...]
      password* > [...]
      about > [...]
      role > [...]
      enabled > [...]
      imageURL > [...]
      phone > [...]
      address > [...]
      posts > [...]
      doctorsAdditionalInfo
        DoctorsAdditionalInfo < {
          id > [...]
          nid > [...]
          doctortype > [...]
          degrees > [...]
          medicalcollege > [...]
          appointmentfee > [...]
          user
            User < {
              id > [...]
              name > [...]
              email* > [...]
              password* > [...]
              about > [...]
              role > [...]
              enabled > [...]
              imageURL > [...]
              phone > [...]
              address > [...]
              posts > [...]
              doctorsAdditionalInfo > {...}
              doctorsSchedule > {...}
              accountActiveToken > {...}
              dob > [...]
            }
          }
        }
      doctorsSchedule > {...}
      accountActiveToken > {...}
      dob > [...]
    }
  }
}

```

```

DoctorsSchedule < {
  id > [...]
  is_10_00 > [...]
  is_10_30 > [...]
  is_11_00 > [...]
  is_11_30 > [...]
  is_12_00 > [...]
  is_2_00 > [...]
  is_2_30 > [...]
  is_3_00 > [...]
  is_3_30 > [...]
  user > {...}
}

```

```
Likes ▾ {  
  id > [...]  
  likerId > [...]  
  post Posts > {...}  
}
```

```
Posts ▾ {  
  id > [...]  
  postDate > [...]  
  postTime > [...]  
  coverPhoto > [...]  
  likesList > [...]  
  commentsList > [...]  
  savedPostsList > [...]  
  user User > {...}  
  postContent > [...]  
}
```

```
SavedPosts ▾ {  
  id > [...]  
  saverId > [...]  
  posts Posts > {...}  
}
```

```

User v {
  id > [...]
  name > [...]
  email* > [...]
  password* > [...]
  about > [...]
  role > [...]
  enabled > [...]
  imageURL > [...]
  phone > [...]
  address > [...]
  posts > [...]
  doctorsAdditionalInfo > {...}
  doctorsSchedule DoctorsSchedule v {
    id > [...]
    is_10_00 > [...]
    is_10_30 > [...]
    is_11_00 > [...]
    is_11_30 > [...]
    is_12_00 > [...]
    is_2_00 > [...]
    is_2_30 > [...]
    is_3_00 > [...]
    is_3_30 > [...]
    user User v {
      id > [...]
      name > [...]
      email* > [...]
      password* > [...]
      about > [...]
      role > [...]
      enabled > [...]
      imageURL > [...]
      phone > [...]
      address > [...]
      posts > [...]
      doctorsAdditionalInfo > {...}
      doctorsSchedule > {...}
      accountActiveToken > {...}
      dob > [...]
    }
  }
  accountActiveToken > {...}
  dob > [...]
}

```

```

DoctorSignup v {
  name > [...]
  email* > [...]
  password* > [...]
  about > [...]
  role > [...]
  enabled > [...]
  imageURL > [...]
  phone > [...]
  address > [...]
  nid > [...]
  doctortype > [...]
  degrees > [...]
  medicalcollege > [...]
  appointmentfee > [...]
  dob > [...]
}

```

```
UnsavePostDTO ▾ {  
  postId  
  unsaverId      > [...]  
}
```

```
SavedPostsDTO ▾ {  
  saverId  
  postId    > [...]  
}
```

```
LikesDTO ▾ {  
  postId    string  
  likerId   string  
}
```

```
CommentsDTO ▾ {  
  postId      string  
  commenterId string  
  commenterName string  
  commenterImage string  
  comment     string  
}
```

```
AppointDoctorDTO ▾ {  
  appointmentTime string($date-time)  
  doctorID        string  
  doctorFee       string  
}
```

```
DoctorGivePrescriptionDTO ▾ {  
  appointmentID string  
  symptoms      string  
  tests         string  
  advice        string  
  medicines     string  
}
```

```
DeletePostDTO ▾ {  
  postId string  
}
```

```
AdminPatientEditDTO ▾ {  
  id          string  
  enabled     boolean  
  name        string  
  page        string  
  email       string  
  phone       string  
  address     string  
  about       string  
  dob         string  
}
```

```
DeleteUserDTO ▾ {  
  userId      string  
}
```

```
AdminDoctorEditDTO ▾ {  
  id          string  
  enabled     boolean  
  name        string  
  page        string  
  email       string  
  phone       string  
  address     string  
  about       string  
  nid         string  
  doctortype  string  
  degrees     string  
  medicalcollege string  
  appointmentfee string  
  dob         string  
}
```

```
EditUserDTO ▾ {  
  pageNo      string  
  userId      string  
}
```