



# SERVER SIDE MODULE

## CONTENTS

This module has the following files:

1. MODULE\_SERVER\_SIDE.docx
2. MODULE\_SERVER\_SIDE.pdf
3. MODULE\_SERVER\_SIDE\_MEDIA.zip

## INTRODUCTION

In this module, you are asked to create a minimum viable product for doctor and department scheduling system. The detail description and tools that you can use will be described below.

## DESCRIPTION OF PROJECT AND TASKS

This module is divided into 2 phases. In the first phase, you will create a REST Api then in the second phase you will create a frontend application with the following provided frameworks:

- Laravel (v11.x)
- React (v18.^ with react-router and axios)
- Vue (v3.^ with vue-router and axios)

### Phase 1 - REST API

For all endpoints, the request header as default sets Content-type: application/json, but some endpoints have specified Content-type header so you have to adjust it.

#### 1. Authentication

You can use **sanctum** for the token management and it must be valid when placed in the Authorization request header as a **Bearer token**.

##### a. Login

Endpoint	/api/v1/auth/login
Method	POST
Description	For user to log into system
Request	Body: username password
Response Success	HTTP Status Code: 200 Body: { "message": "Login success",



	<pre> "token": "2 Qnt2aqHi0EVwCz3rSH0yiFIKMqey38SdkUZntRvf0e507ff", "user": {   "id": 1,   "full_name": "John Doe",   "username": "Admin",   "created_at": "2023-10-14T15:04:33.000000Z" } </pre>
Response Failed	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre> {   "message": "Wrong username or password" } </pre>

### b. Logout

Endpoint	/api/v1/auth/logout
Method	POST
Description	For user to logout
Request	<p>Headers:</p> <p>Authorization: Bearer &lt;token&gt;</p> <p>Body:</p> <p>username</p> <p>password</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre> {   "message": "Logout success" } </pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre> {   "message": "Unauthenticated." } </pre>

## 2. Department

### a. Create New Department

Endpoint	/api/v1/department
Method	POST
Description	For admin to create a department
Request	<p>Headers:</p> <p>Authorization: Bearer &lt;token&gt;</p> <p>Body:</p> <p>Department ID (string;maxlength:10;required)</p> <p>Name (string;required)</p> <p>Description (string;optional)</p>

Response Success	HTTP Status Code: 200 Body: <pre>{   "message": "Department created" }</pre>
Response Invalid Field	Body Status Code 422: <pre>{   "message": "Invalid field",   "errors": {     "departmentID": [       "Department ID field is required."     ],     "name": [       "Department Name field is required."     ]   } }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{   "message": "Unauthenticated." }</pre>

#### b. Update Existing Department

Endpoint	/api/v1/department/:id
Method	PUT
Description	For admin to update existing department by id.
Request	Headers: Authorization: Bearer <token>  Body: Name (string;required) Description (string;optional)
Response Success	HTTP Status Code: 200 Body: <pre>{   "message": "Department modified" }</pre>
Response Invalid Field	Body Status Code 422: <pre>{   "message": "Invalid field",   "errors": {     "name": [       "Department Name field is required."     ]   } }</pre>
Response Not Found	HTTP Status Code: 404 Body: <pre>{   "message": "Department not found" }</pre>
Response	HTTP Status Code: 401 Body:





	<pre> {   "departmentID": "KD01",   "departmentName": "Poli Gigi",   "departmentDesc": "-----",   "createdAt": "2025-01-20 00:34:10",   "updatedAt": null,   "schedules": {     "id": 61,     "doctorName": "Irving Greenfelder",     "scheduleDate": "2025-01-30",     "startTime": "10:00",     "endTime": "12:00",     "created_at": "2025-01-20 19:16:53"   },   },   ... ] </pre>
Response Invalid Params	<p>HTTP Status Code: 422</p> <p>Body:</p> <pre> {   "message": "Invalid field",   "errors": {     "page": [       "The page field must be at least 0."     ],     "size": [       "The size field must be a number."     ]   } } </pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre> {   "message": "Unauthenticated." } </pre>

### 3. Doctor

#### a. Create New Doctor

Endpoint	/api/v1/doctor
Method	POST
Description	For admin to create a doctor
Request	<p>Headers:</p> <p>Authorization: Bearer &lt;token&gt;</p> <p>Body:</p> <p>Doctor ID (string;maxlength:15;required)</p> <p>Name (string;required)</p> <p>Gender (male/female;required)</p> <p>Phone Number (string;required)</p> <p>Address (string;required)</p> <p>Email (string;required)</p> <p>Bio (string;optional)</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p>

	<pre>{   "message": "Doctor created" }</pre>
Response Invalid Field	<p>Body Status Code 422:</p> <pre>{   "message": "Invalid field",   "errors": {     "doctorID": [       "Doctor ID field is required."     ],     "name": [       "Name field is required."     ],     "gender": [       "Gender field is required.   and must be F or M"     ],     "phone": [       "Phone field is required."     ],     "address": [       "Address field is required."     ],     "email": [       "Email field is required."     ]   } }</pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre>{   "message": "Unauthenticated." }</pre>

## b. Update Existing Doctor

Endpoint	/api/v1/doctor/:id
Method	PUT
Description	For admin to update existing doctor by id.
Request	<p>Headers:</p> <p>Authorization: Bearer &lt;token&gt;</p> <p>Body:</p> <p>Name (string;required)</p> <p>Gender (male/female;required)</p> <p>Phone Number (string;required)</p> <p>Address (string;required)</p> <p>Email (string;required)</p> <p>Bio (string;optional)</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre>{   "message": "Doctor modified" }</pre>
Response Invalid Field	<p>Body Status Code 422:</p> <pre>{   "message": "Invalid field",</pre>



	<pre> "errors": {   "name": [     "Name field is required."   ],   "gender": [     "Gender field is required.   and must be F or M"   ],   "phone": [     "Phone field is required."   ],   "address": [     "Address field is required."   ],   "email": [     "Email field is required."   ] } </pre>
Response Not Found	HTTP Status Code: 404 Body: <pre> {   "message": "Doctor not found" } </pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre> {   "message": "Unauthenticated." } </pre>

### c. Delete Existing Doctor

Endpoint	/api/v1/doctor/:id
Method	DELETE
Description	For admin to delete a doctor, doctor can be deleted if there is no schedule attached
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 204
Response Constraint Error	HTTP Status Code: 422 Body: <pre> {   "message": "Doctor cannot deleted" } </pre>
Response Not Found	HTTP Status Code: 404 Body: <pre> {   "message": "Doctor not found" } </pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre> {   "message": "Unauthenticated." } </pre>

## d. Get All Doctors

Endpoint	/api/v1/doctor
Method	GET
Description	For admin to get all doctor data
Request	<p>Headers: Authorization: Bearer &lt;token&gt;</p> <p>Params: page (integer, minimum 0 and default 0) size (integer, minimum 1 and default 10)</p>
Response Success	<p>HTTP Status Code: 200 Body:</p> <pre>{   "page": 0,   "size": 10,   "doctors": [     {       "doctorID": "DOC001",       "doctorName": "Fulan",       "gender": "M",       "phone": "08881288",       "address": "Jl. Sepanjang Kenangan",       "email": "fulan@gmail.com",       "bio": "----",       "createdAt": "2025-01-20 00:34:10",       "updatedAt": null,       "schedules": {         "id": 61,         "departmentName": "Poli Umum",         "scheduleDate": "2025-01-30",         "startTime": "09:00",         "endTime": "15:00",         "created_at": "2025-01-20 19:16:53"       }     },     ...   ] }</pre>
Response Invalid Params	<p>HTTP Status Code: 422 Body:</p> <pre>{   "message": "Invalid field",   "errors": {     "page": [       "The page field must be at least 0."     ],     "size": [       "The size field must be a number."     ]   } }</pre>
Response Invalid Token	<p>HTTP Status Code: 401 Body:</p> <pre>{   "message": "Unauthenticated." }</pre>



#### 4. Schedule

##### a. Create New Schedule

Endpoint	/api/v1/schedule
Method	POST
Description	For admin to create a schedule. Make sure that there is no conflicting schedule (1 doctor with 2 schedules with conflicting hours).
Request	<p>Headers:</p> <p>Authorization: Bearer &lt;token&gt;</p> <p>Body:</p> <p>Doctor ID (string; maxlength:15; required)</p> <p>Department ID (string; maxlength: 10; required)</p> <p>Date (date; min:tomorrow; required)</p> <p>Start Time (time; required)</p> <p>End Time (time; required)</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre>{   "message": "Schedule created" }</pre>
Response Invalid Field	<p>Body Status Code 422:</p> <pre>{   "message": "Invalid field",   "errors": {     "doctorID": [       "Doctor field is required."     ],     "departmentID": [       "Department field is required."     ],     "date": [       "Date field is required.   Minimal tomorrow"     ],     "startTime": [       "Start Time field is required."     ],     "endTime": [       "End Time field is required."     ]   } }</pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre>{   "message": "Unauthenticated." }</pre>

##### b. Update Existing Schedule

Endpoint	/api/v1/schedule/:id
Method	PUT
Description	For admin to update existing schedule by id. The schedule cannot be changed once the date has passed the current date.



Request	<p>Headers: Authorization: Bearer &lt;token&gt;</p> <p>Body: Doctor ID (string; maxlength:15; required) Department ID (string; maxlength: 10; required) Date (date; min:tomorrow; required) Start Time (time; required) End Time (time; required)</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre>{   "message": "Schedule modified" }</pre>
Response Invalid Field	<p>Body Status Code 422:</p> <pre>{   "message": "Invalid field",   "errors": {     "doctorID": [       "Doctor field is required."     ],     "departmentID": [       "Department field is required."     ],     "date": [       "Date field is required.   Minimal tomorrow"     ],     "startTime": [       "Start Time field is required."     ],     "endTime": [       "End Time field is required."     ]   } }</pre>
Response Not Found	<p>HTTP Status Code: 404</p> <p>Body:</p> <pre>{   "message": "Schedule not found" }</pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre>{   "message": "Unauthenticated." }</pre>

### c. Delete Existing Schedule

Endpoint	/api/v1/schedule/:id
Method	DELETE
Description	For admin to delete a schedule. The schedule cannot be deleted once the date has passed the current date.
Request	<p>Headers: Authorization: Bearer &lt;token&gt;</p>
Response	HTTP Status Code: 204



Success	
Response Constraint Error	HTTP Status Code: 422 Body: <pre>{   "message": "Schedule cannot deleted" }</pre>
Response Not Found	HTTP Status Code: 404 Body: <pre>{   "message": "Schedule not found" }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{   "message": "Unauthenticated." }</pre>

#### d. Get All Schedules

Endpoint	/api/v1/schedule
Method	GET
Description	For admin to get all schedules data
Request	Headers: Authorization: Bearer <token>  Params: page (integer, minimum 0 and default 0) size (integer, minimum 1 and default 10)
Response Success	HTTP Status Code: 200 Body: <pre>{   "page": 0,   "size": 10,   "schedule": [     {       "scheduleID": "1",       "doctorID": "DOC000000000001",       "doctorName": "Fulan",       "departmentID": "DEPT000001",       "departmentName": "Poli Umum",       "scheduleDate": "2025-01-30",       "startTime": "09:00",       "endTime": "15:00",       "createdAt": "2025-01-20 00:34:10",       "updatedAt": null     },     ...   ] }</pre>
Response Invalid Params	HTTP Status Code: 422 Body: <pre>{   "message": "Invalid field",   "errors": {     "page": [       "The page field must be at least 0."     ]   } }</pre>



	<pre>    ],     "size": [         "The size field must be a number."     ]   } }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: { "message": "Unauthenticated." }

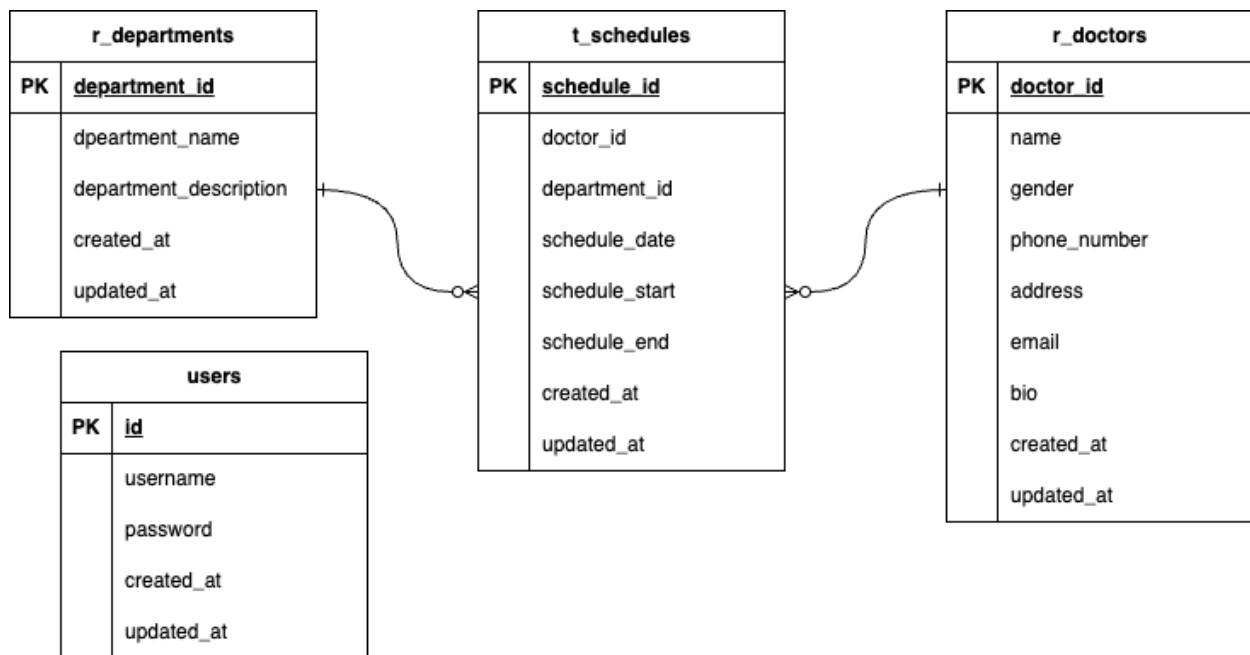


## Phase 2 - Frontend Development

Template HTML is provided. You are required to use the template and you are recommended to modify the template design without affecting the functionality of the frontend.

Page	Description Tasks
<b>General</b>	<ul style="list-style-type: none"> <li>- Document title should be reflected to the current page.</li> <li>- Display username and logout button on the navbar after login success.</li> <li>- Clicking username will go to logged in user profile page.</li> <li>- User can log out after clicking logout button on the navbar.</li> </ul>
<b>Login</b>	<ul style="list-style-type: none"> <li>- If user logged in, the page will be redirected to own user profile page.</li> <li>- User can login with existing credentials.</li> <li>- If login fails, display all of errors message.</li> <li>- If logged in user try to access this page, the page should be redirected. to own user profile page</li> </ul>
<b>Department</b>	<ul style="list-style-type: none"> <li>- On "Department" menu, admin can add new department.</li> <li>- Admin can update and delete existing data, but you cannot do that if there is schedules attached.</li> <li>- <b>Department ID</b> must be a string, no space and consists of 10 Alphanumeric.</li> <li>- <b>Department Name</b> must be a string.</li> <li>- <b>Description</b> must be a string and optional.</li> </ul>
<b>Doctor</b>	<ul style="list-style-type: none"> <li>- On "Doctor" menu, admin can add new doctor</li> <li>- Admin can update and delete existing data, but you cannot do that if there is schedules attached.</li> <li>- <b>Doctor ID</b> must be a string, no space and consists of 15 Alphanumeric.</li> <li>- <b>Name</b> must be a string.</li> <li>- <b>Gender</b> must be a Female or Male with value F or M inserted.</li> <li>- <b>Phone</b> must be a string and maxlength 20.</li> <li>- <b>Address</b> must be a string.</li> <li>- <b>Email</b> must be a string.</li> <li>- <b>Bio</b> must be a string and optional.</li> </ul>
<b>Schedule</b>	<ul style="list-style-type: none"> <li>- On "schedule" menu, admin can add new schedule</li> <li>- Admin can update and delete existing data, but the schedule cannot be changed once the date has passed the current date.</li> <li>- No doctor and department schedule <b>time can't be overlap</b> with other schedule.</li> <li>- The date cannot be applied backwards.</li> <li>- <b>Doctor ID</b> must be a string, consists of 15 Alphanumeric and match with Doctor ID in table r_doctor.</li> <li>- <b>Department ID</b> must be a string, consists of 10 Alphanumeric and match with Department ID in table r_department.</li> <li>- <b>Date</b> must be date and minimum date is tomorrow.</li> <li>- <b>Start Time</b> must be time and minimum time is 6am.</li> <li>- <b>End Time</b> must be time and minimum time is 6am.</li> </ul>

## ER DIAGRAM



## MEDIA FILES

- Frameworks (laravel, react and vue)
- SQL Dump
- Template HTML (based on bootstrap v5.3)

## INSTRUCTION FOR COMPETITORS

- Database (**db-dump.sql**) and ER diagram (**db-diagram.pdf**) should be exported and saved under the desktop folder **XX\_SERVER\_MODULE**
- REST Api should be reached on **<host>/XX\_SERVER\_MODULE/BACKEND** where xx is PC number.

**No port and no /public suffix path.**

- Frontend app should be reached on **<host>/XX\_SERVER\_MODULE/FRONTEND** where xx is PC number.

**No port allowed**

- **Zip XX\_SERVER\_MODULE** and saved under the desktop folder. **Make sure to exclude vendor and node\_modules** folder when zipping files.