SERVER SIDE MODULE PART I

(REST API & AUTHORIZATION)

CONTENTS

This module has the following files:

1. MODULE_SERVER_SIDE_JT1_25.docx
2. MODULE_SERVER_SIDE_JT1_25.pdf
3. MODULE_SERVER_SIDE_MEDIA.zip

INTRODUCTION

In this module, you are asked to create a minimum viable product healthcare application for the specification of the doctor scheduling system. The detailed 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 using Authentification 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 module)
- Vue (v3.^ with vue-router and axios module)

Phase 1 - REST API & Authentication

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/v3/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",
	"token": "2|Qnt2aqHIi0EVwCz3rSH0yiFIKMqey38SdkUZntRvf0e507ff",
	"user": {
	"user_id": 1,
	"user_full_name": "Budiman",
	"user_username": "Admin",
	"created_at": "2023-10-14T15:04:33.000000Z"
	}
	}

Response Failed	HTTP Status Code: 401
	Body:
	{
	"message": "username or password are incorrect"
	}

b. Logout

Endpoint	/api/v3/auth/logout
Method	POST
Description	user to logout
Request	Headers: Authorization: Bearer <token> Body: username password

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Logout success"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

2. Poliklinik

a. Create Poliklinik

Endpoint	/api/v3/poliklinik
Method	POST
Description	Admin can create a Poliklinik

Request	Headers: Authorization: Bearer <token> Body: Poliklinik ID (string;maxlength:10;required) Name (string;required) Description (string;optional)

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Poliklinik is created"
	}

Response Invalid Field	Body Status Code 422:
	{
	"message": "Invalid field",
	"errors": {
	"PoliklinikID": [
	"Poliklinik ID field is required."
	],
	"name": [
	"Poliklinik Name field is required."
	]
	}
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

b. Update Poliklinik

Endpoint	/api/v3/Poliklinik/:id
Method	PUT
Description	Admin can update the Poliklinik by id.
Request	Headers: Authorization: Bearer <token> Body: Name (string;required) Description (string;optional)

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Poliklinik modified"
	}

Response Invalid Field	Body Status Code 422:
	{
	"message": "Invalid field",
	"errors": {
	"name": [
	"Poliklinik Name field is required."
	]
	}

	}

Response Not Found	HTTP Status Code: 404
	Body:
	{
	"message": "Poliklinik not found"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

c. Delete Poliklinik

Endpoint	/api/v3/Poliklinik/:id
Method	DELETE
Description	Admin can delete the Poliklinik 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:
	{
	"message": "Poliklinik cannot deleted"
	}

Response Not Found	HTTP Status Code: 404
	Body:
	{
	"message": "Poliklinik not found"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

d. Get All Poliklinik

Endpoint	/api/v3/Poliklinik
Method	GET
Description	Admin to get all Poliklinik 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:
	{
	"page": 0,
	"size": 10,
	"Polikliniks": [
	{
	"PoliklinikID": "KD01",
	"PoliklinikName": "Poli Gigi",
	"PoliklinikDesc": "-----",
	"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"
	},
	},
	….
	]
	}

Response Invalid Params	HTTP Status Code: 422
	Body:
	{
	"message": "Invalid field",
	"errors": {
	"page": [
	"The page field must be at least 0."
	],
	"size": [
	"The size field must be a number."
	]
	}
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

3. Doctor

a. Create Doctor

Endpoint	/api/v3/doctor
Method	POST
Description	Admin to create a doctor
Request	Headers: Authorization: Bearer <token> Body: Doctor ID (string;maxlength:15;required)

	Name (string;required) Gender (male/female;required) Phone Number (string;required) Address (string;required) Email (string;required) Bio (string;optional)

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Doctor created"
	}

Response Invalid Field	Body Status Code 422:
	{
	"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."
	]
	}
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

b. Update Doctor

Endpoint	/api/v3/doctor/:id
Method	PUT
Description	Admin can update existing doctors by id.
Request	Headers: Authorization: Bearer <token> Body: Name (string;required) Gender (male/female;required) Phone Number (string;required) Address (string;required) Email (string;required) Bio (string;optional)

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Doctor modified"
	}

Response Invalid Field	Body Status Code 422:
	{
	"message": "Invalid field",
	"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."
	]
	}
	}

Response Not Found	HTTP Status Code: 404
	Body:
	{
	"message": "Doctor not found"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

c. Delete Doctor

Endpoint	/api/v3/doctor/:id
Method	DELETE
Description	Admin can delete doctor transaction 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:
	{
	"message": "Doctor cannot deleted"
	}

Response	HTTP Status Code: 404

Not Found	Body:
	{
	"message": "Doctor not found"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

d. Get All Doctors

Endpoint	/api/v3/doctor
Method	GET
Description	Admin to get all doctor 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:
	{
	"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,
	"PoliklinikName": "Poli Umum",
	"scheduleDate": "2025-01-30",
	"startTime": "09:00"
	"endTime ": "15:00",
	"created_at": "2025-01-20 19:16:53"
	},
	},
	….
	]
	}

Response Invalid Params	HTTP Status Code: 422
	Body:
	{
	"message": "Invalid field",
	"errors": {
	"page": [
	"The page field must be at least 0."
	],
	"size": [

	"The size field must be a number."] } }

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

4. Schedule

a. Create Schedule

Endpoint	/api/v3/schedule
Method	POST
Description	Admin to create a schedule. Make sure that there is no conflicting schedule (1 doctor with 2 schedules with conflicting hours).
Request	Headers: Authorization: Bearer <token> Body: Doctor ID (string; maxlength:15; required) Poliklinik ID (string; maxlength: 10; required) Date (date; min:tomorrow; required) Start Time (time; required) End Time (time; required)

Response Success	HTTP Status Code: 200
	Body:
	{
	"message": "Schedule created"
	}

Response Invalid Field	Body Status Code 422:
	{
	"message": "Invalid field",
	"errors": {
	"doctorID": [
	"Doctor field is required."
	],
	"PoliklinikID": [
	"Poliklinik field is required."
	],
	"date": [
	"Date field is required. | Minimal tomorrow"
	],
	"startTime": [
	"Start Time field is required."
	],
	"endTime": [
	"End Time field is required."
	]
	}
	}

Response Invalid Token	HTTP Status Code: 401 Body: {

	<pre>{
	    "message": "Unauthenticated."
	}</pre>

b. Update Schedule

Endpoint	/api/v3/schedule/:id
Method	PUT
Description	Admin can update existing schedules by id. The schedule cannot be changed once the date has passed the current date.
Request	Headers: Authorization: Bearer <token> Body: Doctor ID (string; maxlength:15; required) Poliklinik ID (string; maxlength: 10; required) Date (date; min:tomorrow; required) Start Time (time; required) End Time (time; required)

Response Success	HTTP Status Code: 200
	Body:
	<pre>{
	    "message": "Schedule modified"
	}</pre>

Response Invalid Field	Body Status Code 422:
	<pre>{
	    "message": "Invalid field",
	    "errors": {
	        "doctorID": [
	            "Doctor field is required."
	        ],
	        "PoliklinikID": [
	            "Poliklinik 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	HTTP Status Code: 404
	Body:
	<pre>{
	    "message": "Schedule not found"
	}</pre>

Response Invalid	HTTP Status Code: 401 Body: <pre>{</pre>

Token	"message": "Unauthenticated." }

c. Delete Schedule

Endpoint	/api/v3/schedule/:id
Method	DELETE
Description	Admin can delete a schedule. The schedule cannot be deleted once the date has passed the current date.
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 204

Response Constraint Error	HTTP Status Code: 422
	Body:
	{
	"message": "Schedule cannot deleted"
	}

Response Not Found	HTTP Status Code: 404
	Body:
	{
	"message": "Schedule not found"
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

d. Get All Schedules

Endpoint	/api/v3/schedule
Method	GET
Description	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: { "page": 0, "size": 10,

	"schedule": [
	{
	"scheduleID": "1",
	"doctorID": "DOC00000000001",
	"doctorName": "Fulan",
	"PoliklinikID": "DEPT000001",
	"PoliklinikName": "Poli Umum",
	"scheduleDate": "2025-01-30",
	"startTime": "09:00"
	"endTime ": "15:00",
	"createdAt": "2025-01-20 00:34:10",
	"updatedAt": null
	},
	….
	]
	}

Response Invalid Params	HTTP Status Code: 422
	Body:
	{
	"message": "Invalid field",
	"errors": {
	"page": [
	"The page field must be at least 0."
	],
	"size": [
	"The size field must be a number."
	]
	}
	}

Response Invalid Token	HTTP Status Code: 401
	Body:
	{
	"message": "Unauthenticated."
	}

SERVER SIDE MODULE PART II

(FRONTEND DEVELOPMENT)

CONTENTS

This module has the following files:

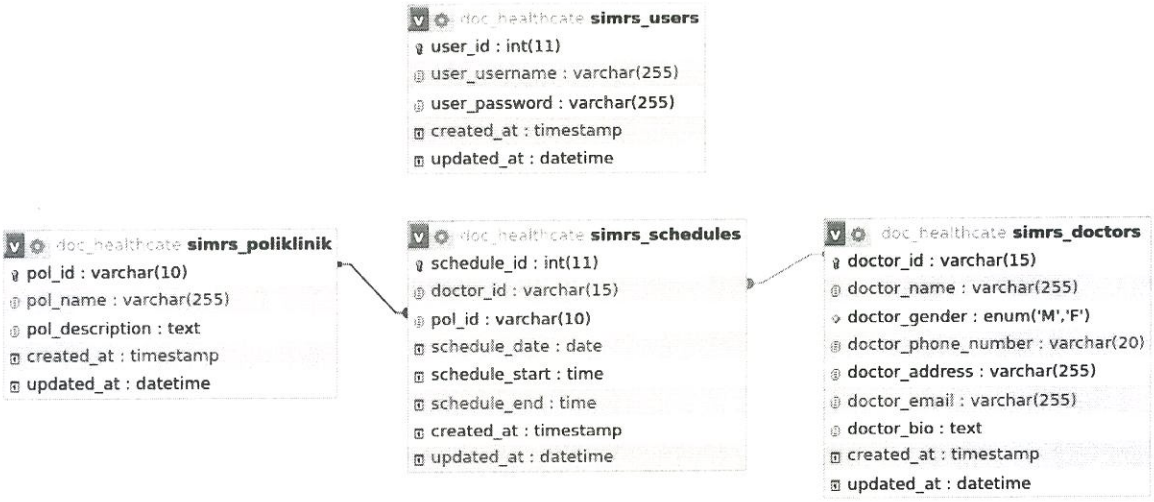
1. MODULE_SERVER_SIDE_JT1_25.docx
2. MODULE_SERVER_SIDE_JT1_25.pdf
3. MODULE_SERVER_SIDE_MEDIA.zip

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
General	- Document title should be reflected on the current page. - Display username and logout button on the navbar after login success. - Clicking username will go to the logged in user profile page. - Users can log out after clicking the logout button on the navbar.
Login	- If a user is logged in, the page will be redirected to its own user profile page. - Users can login with existing credentials. - If login fails, display all of the error messages. - If logged in user try to access this page, the page should be redirected. to own user profile page
Poliklinik	- On the "Poliklinik" menu, the admin can add a new Poliklinik. - Admin can update and delete existing data, but you cannot do that if there are schedules attached. - **Poliklinik ID** must be a string, no space and consists of 10 Alphanumeric. - **Poliklinik Name** must be a string. - **Description** must be a string and optional.
Doctor	- On "Doctor" menu, admin can add new doctor - Admin can update and delete existing data, but you cannot do that if there are schedules attached. - **Doctor ID** must be a string, no space and consists of 15 Alphanumerics. - **Name** must be a string. - **Gender** must be a Female or Male with value F or M inserted. - **Phone** must be a string and maxlength 20. - **Address** must be a string. - **Email** must be a string. - **Bio** must be a string and optional.
Schedule	- On "schedule" menu, admin can add new schedule - Admin can update and delete existing data, but the schedule cannot be

	changed once the date has passed the current date. - No doctor and Poliklinik schedule **time can't overlap** with other schedules. - The date cannot be applied backwards. - **Doctor ID** must be a string, consists of 15 Alphanumeric and match with Doctor ID in table r_doctor. - **Poliklinik ID** must be a string, consists of 10 Alphanumeric and match with Poliklinik ID in table r_Poliklinik. - **Date** must be date and minimum date is tomorrow. - **Start Time** must be time and minimum time is 6am. - **End Time** must be time and minimum time is 6am.

ER DIAGRAM



MEDIA FILES

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

INSTRUCTION FOR COMPETITORS

- Database (**db-healthcare.sql**) and ER diagram (**db-diagram.pdf**) should be exported and saved under the desktop folder **XX_SERVER_MODULE** (Backend / Frontend)
- REST Api should be reached on **<host>/XX/backed** where xx is PC number.
- Frontend app should be reached on **<host>/XX/frontend** where xx is PC number.
- **Zip XX_SERVER_MODULE** and saved under the desktop folder. **Make sure to exclude vendor** and **node_modules** folder when zipping files