



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 social media website called “Facegram” where in the website a user can see other user's posts by following the user. Users can also register as a private account so other users who are not following could not see posts. 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 then in the second phase you will create a frontend application with the following provided frameworks:

- Laravel (v10.x)
- React (v18.x with react-router and axios)
- Vue (v3.x 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. Register

Endpoint	/api/v1/auth/register
Method	POST
Description	For user to register account
Request	Body: full_name (required) bio (required, max 100 chars) username (required, min 3 chars, unique, only alphanumeric, dot “.” or underscore “_”

	allowed) password (required, min 6 chars) is_private (boolean)
Response Success	HTTP Status Code: 201 Body: <pre>{ "message": "Register success", "token": "7 xuPEGo7jtV8IhoE2Mp3amlrwtIpYUTewXBTcH78Af922f116", "user": { "full_name": "Budi Budiman", "bio": "stop dreaming, start doing", "username": "budi.budiman", "is_private": true, "id": 101 } }</pre>
Response Invalid Field	HTTP Status Code: 422 Body: <pre>{ "message": "Invalid field", "errors": { "full_name": ["The full name field is required."], "username": ["The username has already been taken."], "password": ["The password field must be at least 6 characters."], "bio": ["The bio field is required."] } }</pre>

b. 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: <pre>{ "message": "Login success", "token": "2 Qnt2aqHIi0EVwCz3rSH0yiFIKMqey38SdkUZntRvf0e507ff", "user": { "id": 1, "full_name": "John Doe", "username": "john.doe", "bio": "The best way to predict the future is to create it.", "is_private": 0, "created_at": "2023-10-14T15:04:33.000000Z" } }</pre>



Response Failed	HTTP Status Code: 401 Body: <pre>{ "message": "Wrong username or password" }</pre>
-----------------	--

c. Logout

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

2. Post

a. Create new post

Endpoint	/api/v1/posts
Method	POST
Description	For user to create a post
Request	Headers: Authorization: Bearer <token> Content-type: multipart/form-data Body: caption (required) attachments (required, array of image files jpg, jpeg, webp, png or gif)
Response Success	HTTP Status Code: 201 Body: <pre>{ "message": "Create post success" }</pre>
Response Invalid Field	Body: <pre>{ "message": "Invalid field", }</pre>



	<pre> "errors": { "caption": ["The caption field is required."], "attachments.0": ["The attachments.0 field must be a file of type: png, jpg, jpeg, webp."] } </pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre> { "message": "Unauthenticated." } </pre>

b. Delete post

Endpoint	/api/v1/posts/:id
Method	DELETE
Description	For user to delete a post
Request	<p>Headers:</p> <p>Authorization: Bearer <token></p>
Response Success	HTTP Status Code: 204
Response Post Not Found	<p>HTTP Status Code: 404</p> <p>Body:</p> <pre> { "message": "Post not found" } </pre>
Response Try to delete another user's post	<p>HTTP Status Code: 403</p> <p>Body:</p> <pre> { "message": "Forbidden access" } </pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre> { "message": "Unauthenticated." } </pre>



c. Get posts

Endpoint	/api/v1/posts
Method	GET
Description	For user to get all posts
Request	<p>Headers:</p> <p>Authorization: Bearer <token></p> <p>Params:</p> <p>page (integer, minimum 0 and default 0)</p> <p>size (integer, minimum 1 and default 10)</p>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre>{ "page": 0, "size": 10, "posts": [{ "id": 366, "caption": "Letting my soul wander", "created_at": "2023-07-26 00:34:10", "deleted_at": null, "user": { "id": 61, "full_name": "Irving Greenfelder", "username": "talial94", "bio": "Success is not final, failure is not fatal: It is the courage to continue that counts.", "is_private": 0, "created_at": "2023-10-18 19:16:53" }, "attachments": [{ "id": 750, "storage_path": "posts/652fc6284eb76.jpg" }] }, ...] }</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 Try to delete another user's post	<p>HTTP Status Code: 403</p> <p>Body:</p> <pre>{ "message": "Forbidden access" }</pre>



Response Invalid Token	HTTP Status Code: 401 Body: <pre>{ "message": "Unauthenticated." }</pre>
------------------------	--

3. Following

a. Follow a user

Endpoint	/api/v1/users/:username/follow
Method	POST
Description	For user to follow another user
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 200 Body: <pre>{ "message": "Follow success", "status": "following" "requested" }</pre>
Response User not found	HTTP Status Code: 404 Body: <pre>{ "message": "User not found" }</pre>
Response Try to follow own user account	HTTP Status Code: 422 Body: <pre>{ "message": "You are not allowed to follow yourself" }</pre>
Response Already Followed	HTTP Status Code: 422 Body: <pre>{ "message": "You are already followed", "status": "following" "requested" }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{ "message": "Unauthenticated." }</pre>



b. Unfollow a user

Endpoint	/api/v1/users/:username/unfollow
Method	DELETE
Description	For user to unfollow a followed user
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 204
Response User not found	HTTP Status Code: 404 Body: { "message": "User not found" }
Response Not Following the User	HTTP Status Code: 422 Body: { "message": "You are not following the user" }
Response Invalid Token	HTTP Status Code: 401 Body: { "message": "Unauthenticated." }



c. Get following users

Endpoint	/api/v1/following
Method	GET
Description	For users to get their following users
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 200 Body: <pre>{ "following": [{ "id": 91, "full_name": "Miss Ayana Russel", "username": "chartmann", "bio": "In a world where you can be anything, be kind.", "is_private": 0, "created_at": "2023-10-18 19:16:55", "is_requested": true false }, ...] }</pre>
Response User not found	HTTP Status Code: 404 Body: <pre>{ "message": "User not found" }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{ "message": "Unauthenticated." }</pre>

4. Followers

a. Accept follow request

Endpoint	/api/v1/users/:username/accept
Method	PUT
Description	For users to get their following users
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 200 Body: { "message": "Follow request accepted" }
Response User not found	HTTP Status Code: 404 Body: { "message": "User not found" }
Response Already Accepted	HTTP Status Code: 422 Body: { "message": "Follow request is already accepted" }
Response User Not Following	HTTP Status Code: 422 Body: { "message": "The user is not following you" }
Response Invalid Token	HTTP Status Code: 401 Body: { "message": "Unauthenticated." }



b. Get follower users

Endpoint	/api/v1/users/:username/followers
Method	GET
Description	For users to get their following users
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 200 Body: <pre>{ "followers": [{ "id": 91, "full_name": "Miss Ayana Russel", "username": "chartmann", "bio": "In a world where you can be anything, be kind.", "is_private": 0, "created_at": "2023-10-18 19:16:55", "is_requested": true false }, ...] }</pre>
Response User not found	HTTP Status Code: 404 Body: <pre>{ "message": "User not found" }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{ "message": "Unauthenticated." }</pre>



5. User

a. Get users

Endpoint	/api/v1/users
Method	GET
Description	Get all users that are not followed by logged in users. Logged in users should be hidden in the list.
Request	Headers: Authorization: Bearer <token>
Response Success	HTTP Status Code: 200 Body: <pre>{ "users": [{ "id": 1, "full_name": "John Doe", "username": "john.doe", "bio": "The best way to predict the future is to create it.", "is_private": 0, "created_at": "2023-10-14T15:04:33.000000Z", "updated_at": "2023-10-14T15:04:33.000000Z" }, ...] }</pre>
Response Invalid Token	HTTP Status Code: 401 Body: <pre>{ "message": "Unauthenticated." }</pre>



b. Get detail user

Endpoint	/api/v1/users/:username
Method	GET
Description	For users to get detailed user data. Field posts should be hidden if the user is a private user and the follow status is not following or is requested.
Request	Headers: Authorization: Bearer <token>
Response Success	<p>HTTP Status Code: 200</p> <p>Body:</p> <pre>{ "id": 2, "full_name": "Richard Roe", "username": "richard.roe", "bio": "Leave a little sparkle wherever you go.", "is_private": 1, "created_at": "2023-10-18 19:16:50", "is_your_account": false, "following_status": "not-following" "requested" "following", "posts_count": 6, "followers_count": 20, "following_count": 18, "posts": [{ "id": 7, "caption": "Blissfully lost in the beauty of the moment.", "created_at": "2014-08-26 10:42:31", "deleted_at": null, "attachments": [{ "id": 13, "storage_path": "posts/shutterstock_1464930743-scaled.webp" }, ...] }, ...] }</pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre>{ "message": "Unauthenticated." }</pre>
Response User not found	<p>HTTP Status Code: 404</p> <p>Body:</p> <pre>{ "message": "User not found" }</pre>
Response Invalid Token	<p>HTTP Status Code: 401</p> <p>Body:</p> <pre>{ "message": "Unauthenticated." }</pre>



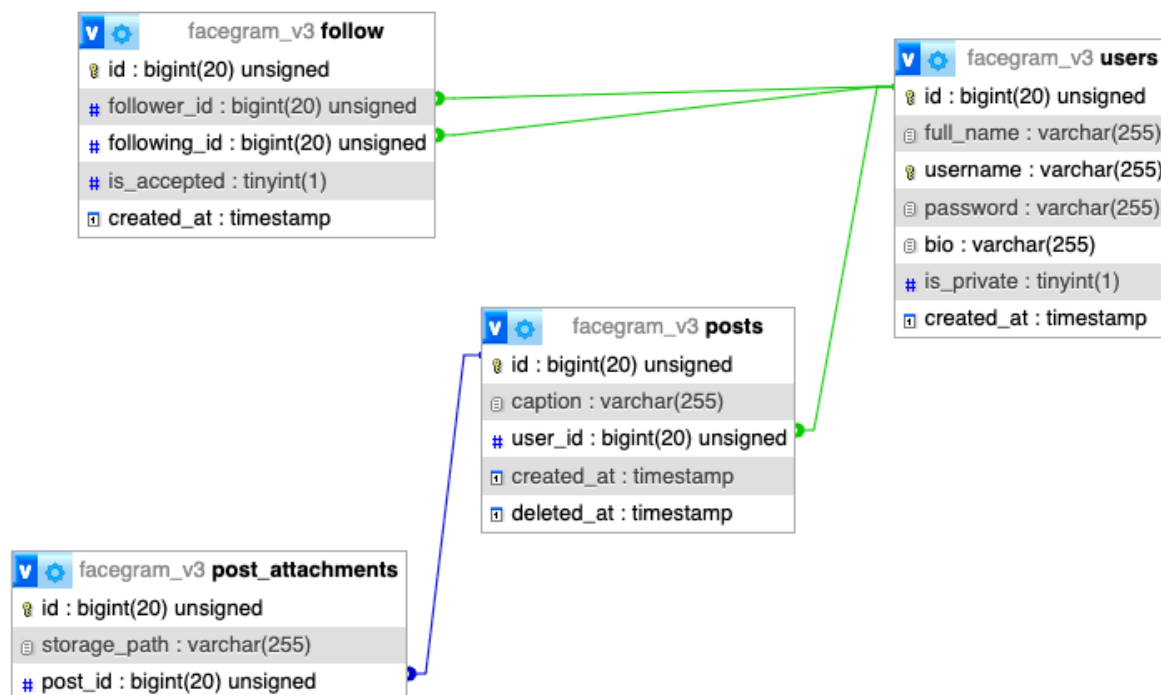
Phase 2 - Frontend Development

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.

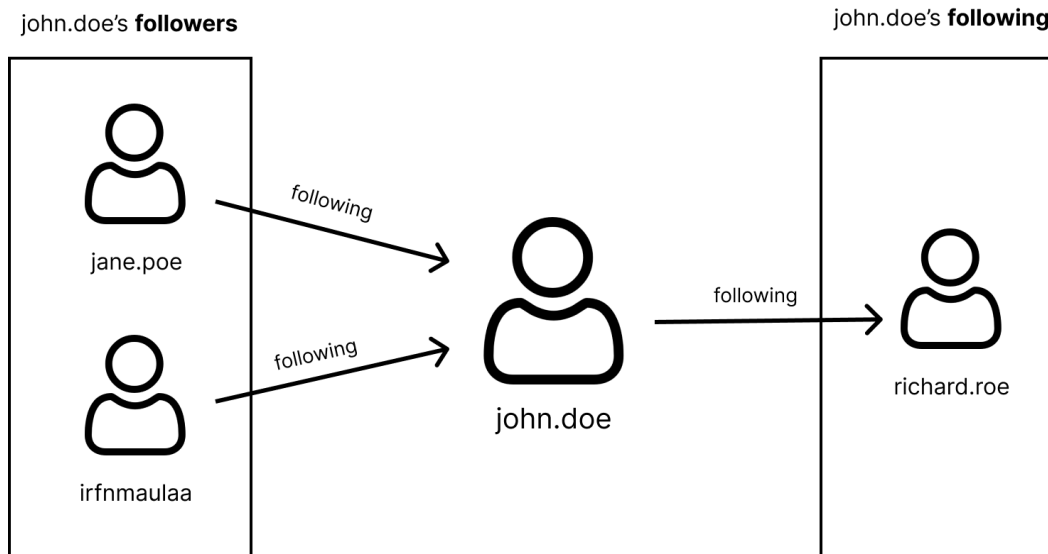
Page	Description Tasks
General	<ul style="list-style-type: none"> - Document title should be reflected to the current page - Display username and logout button on the navbar after login success - Clicking username will go to logged in user profile page - User can log out after clicking logout button on the navbar
Register	<ul style="list-style-type: none"> - If user logged in, the page will be redirected to own user profile page - User can register account - If register fails, display all of errors message - If logged in user try to access this page, the page should be redirected to own user profile page
Login	<ul style="list-style-type: none"> - If user logged in, the page will be redirected to own user profile page - User can login with existing credentials - If login fails, display all of errors message - If logged in user try to access this page, the page should be redirected to own user profile page
Homepage	<ul style="list-style-type: none"> - If user not logged in, the page will be redirected to login page <p>News Feed section</p> <ul style="list-style-type: none"> - Display following user's posts and own posts sorted by newest post on the newsfeed section - When page loads, display only first 10 newest posts - If the vertical scroll is stuck at the bottom, load the next 7 posts <p>Explore people section</p> <ul style="list-style-type: none"> - Display all users that are not followed by the logged in user - Clicking on username will go to that user profile page <p>Follow requests section (for private account)</p> <ul style="list-style-type: none"> - Display follow request users - The logged in user can accept/approve a follow request - If the logged in user is not a private account or there is no any follow request, hide this section
User profile	<p>Page requirements:</p> <ul style="list-style-type: none"> - If user not logged in, the page will be redirected to login page <p>User information:</p> <ul style="list-style-type: none"> - See profile (full_name, username, bio) - See posts list and count - See followers list and count - See following list and count <p>Follow/unfollow:</p> <ul style="list-style-type: none"> - User can follow another user - User can unfollow a followed user <p>Posts section:</p>

	<ul style="list-style-type: none"> - Display user's posts where the visited user is not a private account or following account - Users can delete their posts - Display "The account is private" if logged in user is not following the visited private user profile page
Create new post	<ul style="list-style-type: none"> - If user not logged in, the page will be redirected to login page - User can create new post with following fields: - caption (text, required) - attachments (file, can attach multiple files) - If create new post fails, display all of errors message

ER DIAGRAM



ADDITIONAL EXPLANATION



MEDIA FILES

- Frameworks (laravel, react and vue)
- SQL File (structure and records. allowed to modify)
- Postman collection
- 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 root folder **<host>/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 submit to the submission page. **Make sure to exclude vendor and node_modules** folder when zipping files.