

**Test Project**

Web Technologies

Module A – Backend Development

## Introduction

Indonesia is a country known for its very diverse culinary delights. many food recipes with high taste prepared by Indonesian people. An Indonesian food company called **“IndoRecipe”** asked you to create an application about **food recipes** where each **user can share, rate and comment** on food recipes.

## Description of projects

In this module, your task is to create a **REST Api** using one of the provided **PHP Frameworks**. There are 5 group endpoints that you should create. All of the response data is **JSON format**.

* Discover categories and recipes (for public)
* Authentication
* Category management (only for admin)
* Recipe management (only for logged in user)
* Rate and comment recipe (only for logged in user)

## a. Discover categories and recipes

There are 4 endpoints that you have to create (get all categories, get all recipes, detail recipe and get top best recipes). Those endpoints can be accessed by guest and logged in users.

### a1. Get all categories

Endpoint: [HOST]/api/v1/categories

Method: GET

Description:

* This endpoint should return all of the existing categories and sorted by name ascending.

Response:

* + Response code: 200
  + Response body:

{

"categories": [

{

"id": 2,

"name": "Appetizer",

"slug": "appetizer"

},

{

"id": 6,

"name": "Condiment",

"slug": "condiment"

},

{

"id": 5,

"name": "Dessert",

"slug": "dessert"

},

{

"id": 1,

"name": "Main Dish",

"slug": "main-dish"

},

…

]

}

### a2. Get all recipes

Endpoint: [HOST]/api/v1/recipes

Method: GET

Description:

* This endpoint should return all of the existing recipes and sorted by created\_at descending.

Response:

* + Response code: 200
  + Response body:

{

"recipes": [

{

"id": 1,

"title": "Nasi Goreng",

"slug": "nasi-goreng",

"ingredients": "Beef or chicken...",

"method": "Boil or steam potato...",

"tips": "Use cold leftover rice...",

"energy": "300 kcal",

"carbohydrate": "45g",

"protein": "10g",

"thumbnail": <VALID\_IMAGE\_URL>,

"created\_at": "2023-08-11 10:56:05",

"author": "user39",

"ratings\_avg": 2.7,

"category": {

"id": 1,

"name": "Main Dish",

"slug": "main-dish"

}

},

...

]

}

### a3. Detail recipe

Endpoint: [HOST]/api/v1/recipes/{slug}

Method: GET

Description:

* This endpoint should return a single recipe data based slug parameter.

Response:

* + Response code: 200
  + Response body:

{

"id": 9,

"title": "Sate Padang",

"slug": "sate-padang",

"ingredients": "Beef or chicken...",

"method": "Boil or steam potato...",

"tips": "Use cold leftover rice...",

"energy": "350 kcal",

"carbohydrate": "10g",

"protein": "25g",

"thumbnail": <VALID\_IMAGE\_URL>,

"created\_at": "2023-07-20 09:33:57",

"author": "user11",

"ratings\_avg": 2.4,

"category": {

"id": 1,

"name": "Main Dish",

"slug": "main-dish"

},

"comments": [

{

"id": 44,

"comment": "Food was so delicious…",

"created\_at": "2023-08-14 04:01:20",

"comment\_author": "user46"

},

...

]

}

### a4. Get top best recipes

Endpoint: [HOST]/api/v1/best-recipes/

Method: GET

Description:

* This endpoint should return 3 recipes with the highest average rating.

Response:

* + Response code: 200
  + Response body:

{

"best\_recipes": [

{

"id": 1,

"title": "Nasi Goreng",

"slug": "nasi-goreng",

"ingredients": "Beef or chicken...",

"method": "Boil or steam potato...",

"tips": "Use cold leftover rice...",

"energy": "300 kcal",

"carbohydrate": "45g",

"protein": "10g",

"thumbnail": <VALID\_IMAGE\_URL>,

"created\_at": "2023-08-11 10:56:05",

"author": "user39",

"ratings\_avg": 2.7,

"category": {

"id": 1,

"name": "Main Dish",

"slug": "main-dish"

}

},

...

]

}

## b. Authentication

There are 4 endpoints that you have to create (register, login, logout and get profile). Logout and get profile endpoint can only be accessed by logged in users.

### b1. Register

Endpoint: [HOST]/api/v1/register

Method: POST

Description:

* For guest users to register themselves to the system. The access token should be generated by **UUID**. This endpoint should register the user with the “User” role, not the Administrator.

Request:

* Request body:
  + username: required, unique
  + password: required, min:6
  + password\_confirmation: required, match with password

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Register success",

"accessToken": "<UUID>"

}

* If invalid field:
  + Response code: 422
  + Response body:

{

"message": "Invalid field”,

"errors": "<ERRORS>"

}

### b2. Login

Endpoint: [HOST]/api/v1/login

Method: POST

Description:

* For guest users to sign into the system with existing credentials. The access token should be generated by **UUID**.

Request:

* Request body:
  + username: required
  + password: required

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Login success",

"role": "Administrator",

"accessToken": "<UUID>"

}

* If failed:
  + Response code: 401
  + Response body:

{

"message": "Username or password incorrect”

}

### b3. Logout

Endpoint: [HOST]/api/v1/logout

Method: POST

Description:

* For logged in users to sign out from the system

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Logout success"

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

### b4. Get profile

Endpoint: [HOST]/api/v1/profile

Method: GET

Description:

* For guest users to get their profile data (username and their created recipes).

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”
* Request body:
  + name: required

Response:

* If success:
  + Response code: 200
  + Response body:

{

"id": 3,

"username": "user3",

"role": "User",

"recipes": [

{

"id": 1,

"title": "Nasi Goreng",

"slug": "nasi-goreng",

"ingredients": "Beef or chicken...",

"method": "Boil or steam potato...",

"tips": "Use cold leftover rice...",

"energy": "300 kcal",

"carbohydrate": "45g",

"protein": "10g",

"thumbnail": <VALID\_IMAGE\_URL>,

"created\_at": "2023-08-11 10:56:05",

"author": "user39",

"ratings\_avg": 2.7,

"category": {

"id": 1,

"name": "Main Dish",

"slug": "main-dish"

}

},

...

]

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

## c. Category Management

There are 2 endpoints that you have to create (create and delete category). Those endpoints can only be accessed by admin.

### c1. Create category

Endpoint: [HOST]/api/v1/categories

Method: POST

Description:

* For admin to create a recipe category

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”
* Request body:
  + name: required
  + slug: required, unique

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Category created successful"

}

* If invalid field:
  + Response code: 422
  + Response body:

{

"message": "Invalid field”,

"errors": "<ERRORS>"

}

* If not admin try to access this endpoint:
  + Response code: 403
  + Response body:

{

"message": "Forbidden access"

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

### c2. Delete category

Endpoint: [HOST]/api/v1/categories/{slug}

Method: DELETE

Description:

* For admin to delete a recipe category

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Category deleted successful"

}

* If not admin try to access this endpoint:
  + Response code: 403
  + Response body:

{

"message": "Forbidden access"

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

## d. Recipe Management

There are 2 endpoints that you have to create (create and delete recipe). Those endpoints can only be accessed by the logged in user (admin or user).

### d1. Create recipe

Endpoint: [HOST]/api/v1/recipes

Method: POST

Description:

* For logged in users to create a recipe

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”
* Request body:
  + title: required
  + category\_id: required, a valid category\_id
  + energy: required, numeric
  + carbohydrate: required, numeric
  + protein: required, numeric
  + ingredients: required
  + method: required
  + tips: required,
  + thumbnail: optional or image png/jpg/jpeg

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Recipe created successful"

}

* If invalid field:
  + Response code: 422
  + Response body:

{

"message": "Invalid field”,

"errors": "<ERRORS>"

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

### d2. Delete recipe

Endpoint: [HOST]/api/v1/recipes/{slug}

Method: DELETE

Description:

* For logged in users to delete a recipe

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Recipe deleted successful"

}

* If invalid field:
  + Response code: 422
  + Response body:

{

"message": "Invalid field”,

"errors": "<ERRORS>"

}

* If a user try to delete other user recipes:
  + Response code: 403
  + Response body:

{

"message": "Forbidden access”

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

## e. Rating and Comment

There are 2 endpoints that you have to create (rating and comment). Those endpoints can only be accessed by the logged in user (admin or user).

### e1. Rating

Endpoint: [HOST]/api/v1/recipes/{slug}/rating

Method: POST

Description:

* For logged in users to rate other user recipes. Users cannot rate recipes they have created and cannot rate a recipe 2 times.

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”
* Request body:
  + rating: required, number between 1 and 5

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Rating success"

}

* If invalid field:
  + Response code: 422
  + Response body:

{

"message": "Invalid field”,

"errors": "<ERRORS>"

}

* If user try to rate own recipe:
  + Response code: 403
  + Response body:

{

"message": "You cannot rate your own recipe”

}

* If user has rated the recipe:
  + Response code: 403
  + Response body:

{

"message": "You have rated”

}

* If invalid token:
  + Response code: 401
  + Response body:

{

"message": "Invalid token”

}

### e2. Comment

Endpoint: [HOST]/api/v1/recipes/{slug}/comment

Method: POST

Description:

* For logged in users to post a comment on a recipe

Request:

* Request headers:
  + Authorization: “Bearer <accessToken>”
* Request body:
  + comment: required

Response:

* If success:
  + Response code: 200
  + Response body:

{

"message": "Comment success"

}

* If invalid token:
  + Response code: 401
  + Response body:

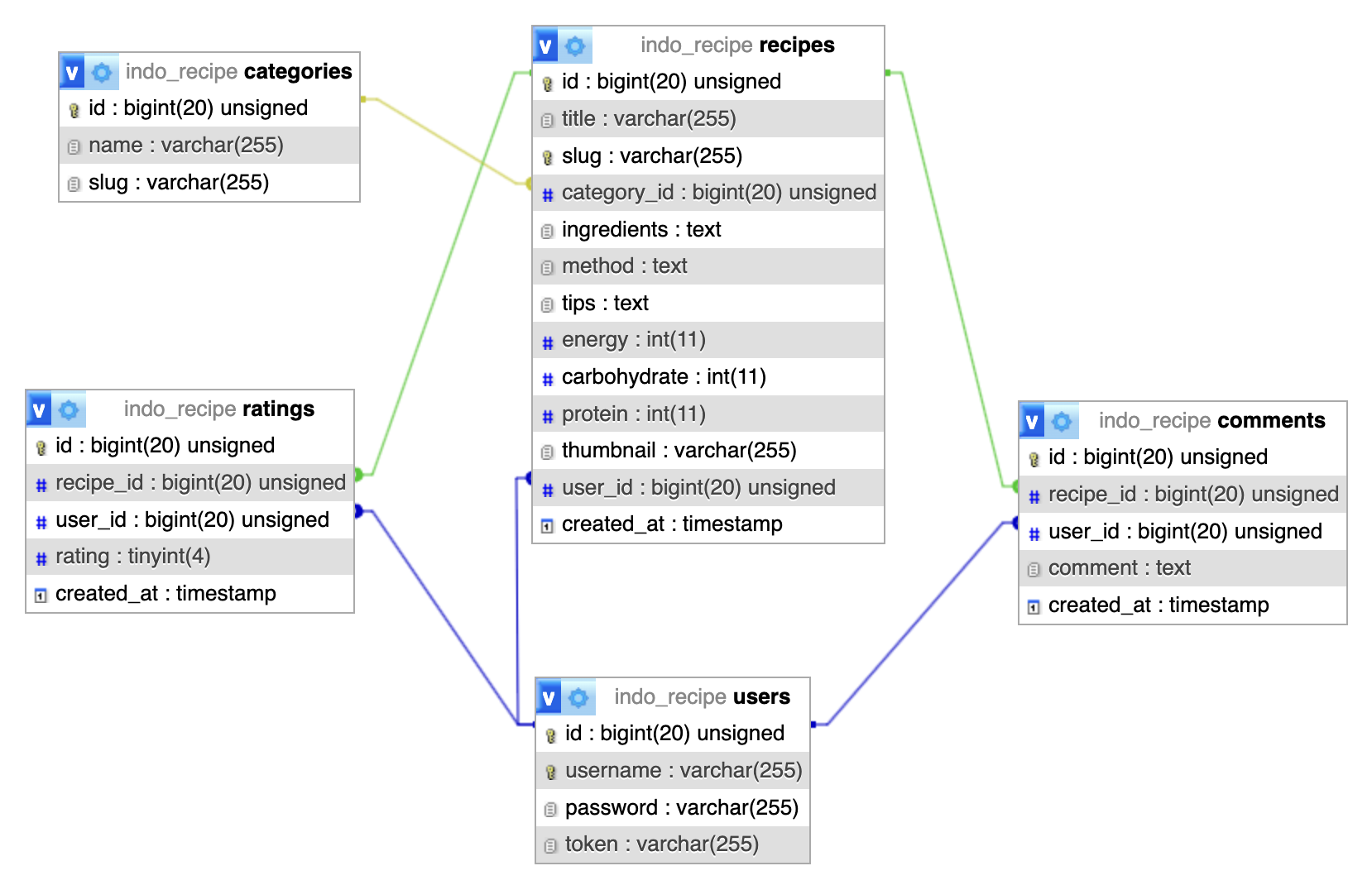
{

"message": "Invalid token”

}

# ERD

You can use ERD below:



# Media Files

## PHP Frameworks (laravel.zip and codeigniter.zip)

## Postman Collection (postman-collection.json)

## SQL File (db-dump.sql)

## User Credentials (credentials.txt)

## Recipe Images

# Instruction For Competitors

## Save your project files to <HOST>/XX\_Module\_Backend where XX is your PC number.

## All routes will be checked under <HOST>/XX\_Module\_Backend/. Note that the path should not have a default /public url.

## Please make sure you have created Administrator account:

| **role** | **username** | **password** |
| --- | --- | --- |
| Administrator | admin | admin |

## Put your database dump file as db-dump.sql and should be able to be downloaded on <HOST>/XX\_Module\_Backend/db-dump.sql

## Put your ERD schema screenshot in your root project folder as db.png should be able to be downloaded on <HOST>/XX\_Module\_Backend/db.png

## For this module, you must use one of the provided PHP frameworks. Applications developed without use of any of these frameworks will not be considered. You should take advantage of the framework as much as possible.

# Marking Summary

| **Sub**  **Criterion**  **ID** | **Sub Criterion**  **Name** | **Measurement Score** | **Judgment**  **Score** | **Total**  **Score** |
| --- | --- | --- | --- | --- |
| A1 | Database | 2.00 | 0 | 2.00 |
| A2 | Discover categories and recipes | 4.25 | 0 | 4.25 |
| A3 | Authentication | 3.50 | 0 | 3.50 |
| A4 | Category Management | 2.75 | 0 | 2.75 |
| A5 | Recipe Management | 4.50 | 0 | 4.50 |
| A6 | Rating and Comment | 5.00 | 0 | 5.00 |
| A7 | General | 1.50 | 1.50 | 3.00 |
| **Total** | | **23.50** | **1.50** | **25.00** |