**REST API MODULE**

# CONTENTS

## This module has the following files:

1. MODULE\_REST\_API.doc
2. MODULE\_REST\_API\_MEDIA.zip

# INTRODUCTION

We all know that nowadays many graduates are unemployed. So the government appoints you as a competent person to create a job seekers platform.

The goal of this platform is so that societies can find work on the platform without having to come to the company location directly and can also see when and where society can come to the company location.

# DESCRIPTION OF PROJECTS AND TASKS

Registration involves the society platform (registering to the platform and applying for jobs), the company platform (giving jobs, viewing and accepting/rejecting registrations from the society), and the officer platform (updating company data, checking the validity of society data).

But your scope of work on this module is only to **create a society platform**. Your job is creating a Society Job application REST API using one of provided PHP frameworks (Laravel).

# Society Job Seekers REST API

## Authentication

Before a society performs any activity on the society platform, they must be logged in using their **ID card number and password**. After successfully logging in, the token will be automatically generated from the **md5 of society ID card number**.

## Specifications :

Feature: A1 - Login and Logout as society: as a visitor, i want to login so that i can enter the system and applying for jobs.

Scenario: Society Login

Request:

URL: [domain]/api/v1/auth/login

Method: POST

Header:

Body: { id\_card\_number: “SomeText”, password: “SomeText” }

Response:

A1a - If login success

Header: Response code: 200

Body: {

"name": "Doni Rianto",

"born\_date": "1974-10-22",

"gender": "male"|”female”,

"address": "Ki. Raya Setiabudhi No. 790",

"token": "e96aaafb6f2f76460b8cc93723bd030e",

"regional": {

"id": 1,

"province": "DKI Jakarta",

"district": "Central Jakarta"

}

}

A1b - If ID Card Number or Password incorrect

Header: Response code: 401

Body: { "message": "ID Card Number or Password incorrect" }

Scenario: Society Logout

Request:

URL: [domain]/api/v1/auth/logout

Params: { token }

Method: POST

Header:

Body:

Response:

A1c - If logout success

Header: Response code: 200

Body: { "message": "Logout success" }

A1d - If logout invalid token

Header: Response code: 401

Body: { "message": "Invalid token" }

# Request Data Validation

To be able to applying for jobs, the society must validation data on the platform first. The society can view validity status and make validation requests. The society can only make one validation data request.

## Specifications:

Feature: A2 - Request Data Validation: As a society, I want to request data validaton so that I can applying for jobs when my data validation is accepted.

Scenario: Request data validations

Request:

URL: [domain]/api/v1/validation

Params: { token }

Method: POST

Header:

Body: { work\_experience: “SomeText”, job\_category: “JobCategoryID”, job\_position: “SomeText” , reason\_accepted: “SomeText” }

Response:

A2a - If request data validation success

Header: Response code: 200

Body: { "message": "Request data validation sent successful" }

A2b - If invalid token

Header: Response code: 401

Body: { "message": "Unauthorized user" }

Scenario: Get society data validation

Request:

URL: [domain]/api/v1/validations

Params: { token }

Method: GET

Header:

Body:

Response:

A2c - If success

Header: Response code: 200

Body: {

“validation”: {

"id": 1,

"status": "pending”|”accepted”|”rejected",

"work\_experience": "SomeText”|null,

"job\_category\_id": "SomeText"|null,

"job\_position": "SomeText"|null,

"reason\_accepted": “SomeText”|null,

"validator\_notes": “SomeText”|null,

"validator": { VALIDATOR }|null

}

}

A2d - If invalid token

Header: Response code: 401

Body: {"message": "Unauthorized user"}

# Job Vacancy

When the validation data has been accepted by the validator and society will applying for jobs, the society can choose the job varancy based on the job category that has been selected during data validation. List of available job varancy at each company should also be displayed.

In the job varancy detail, calculate the number of people who have registered for this job.

**Specifications**:

Feature: A3 – Job Vacancy: As a society, I want to see job vacancy in my job category.

Scenario: Get all job vacancy by choosen job category

Request:

URL: [domain]/api/v1/job\_vacancies

Params: { token }

Method: GET

Header:

Body:

Response:

A3a - If success

Header: Response code: 200

Body: {

"vacancies": [

{

"id": 1,

"category" : {

"id": 1,

"job\_category": "SomeText",

},

"Company": "PT. Maju Mundur Sejahtera",

"address": "Jr. Gotong Royong",

"description": "Lowongan pekerjaan untuk

beberapa list berikut",

"available\_position": [

{

"position": "Web Developer",

"capacity": 2,

"apply\_capacity": 15,

},

...,

]

},

...,

]

}

A3b - If invalid token

Header: Response code: 401

Body: { "message": "Unauthorized user" }

Scenario: Get job vacancy detail by vacancy ID and date

Request:

URL: /api/v1/job\_vacancies/< id>

Params: { token, id }

Method: GET

Header:

Body:

Response:

A3c - If success

Header: Response code: 200

Body: {

"vacancy": {

"id": 1,

"category" : {

"id": 1,

"job\_category": "SomeText",

},

"company": "PT. Maju Mundur Sejahtera",

"address": "Jr. Gotong Royong",

"description": "Lowongan pekerjaan untuk

beberapa list berikut",

"available\_position": [

{

"position": "Web Developer",

"capacity": 2,

"apply\_capacity": 15,

"apply\_count": 10,

},

...,

]

},

}

A3d - If invalid token

Header: Response code: 401

Body: { "message": "Unauthorized user" }

# Applying for Jobs

The society can applying for a jobs if the validation data has been accepted by the validator.

Application for a job vacancy can only be once.

Vacancy ID and position must be filled where the society can choose more than one position where the position has not reached the maximum apply.

## Specifications:

Feature: A4 – Applying for Jos: as a society, i want to applying for jobs.

Scenario: Applying for jobs

Request:

URL: /api/v1/applications

Params: { token }

Method: POST

Header:

Body: { vacancy\_id: 1, positions: [position 1, ...],

notes : “SomeText” }

Response:

A4a - If success

Header: Response code: 200

Body: { "message": "Applying for job successful" }

A4b - If invalid token

Header: Response code: 401

Body: { "message": "Unauthorized user" }

A4c - If the society validation data hasn’t accepted by validator

Header: Response code: 401

Body: { "message": "Your data validator must be accepted by validator before" }

A4d - If invalid fields

Header: Response code: 401

Body: {

"message": "Invalid field",

"errors": {

"vacancy\_id": [

"The vacancy id field is required."

],

"positions ": [

"The position field is required."

]

}

}

A4e - If have been 2x applying

Header: Response code: 401

Body: { "message": " Application for a job can only be once" }

Scenario: Get all of society job applications

Request:

URL: [domain]/api/v1/applications

Params: { token }

Method: GET

Header:

Body: Response:

A4f - If success

Header: Response code: 200

Body: {

"vacancies": [

{

"id": 1,

"category" : {

"id": 1,

"job\_category": "SomeText",

},

"company": "PT. Maju Mundur Sejahtera",

"address": "Jr. Gotong Royong",

"position": [

{

"position": "Web Developer",

"apply\_status": "pending”|”accepted”|”rejected",

"notes": "I want this position",

},

..., ]

},

...,

]

}

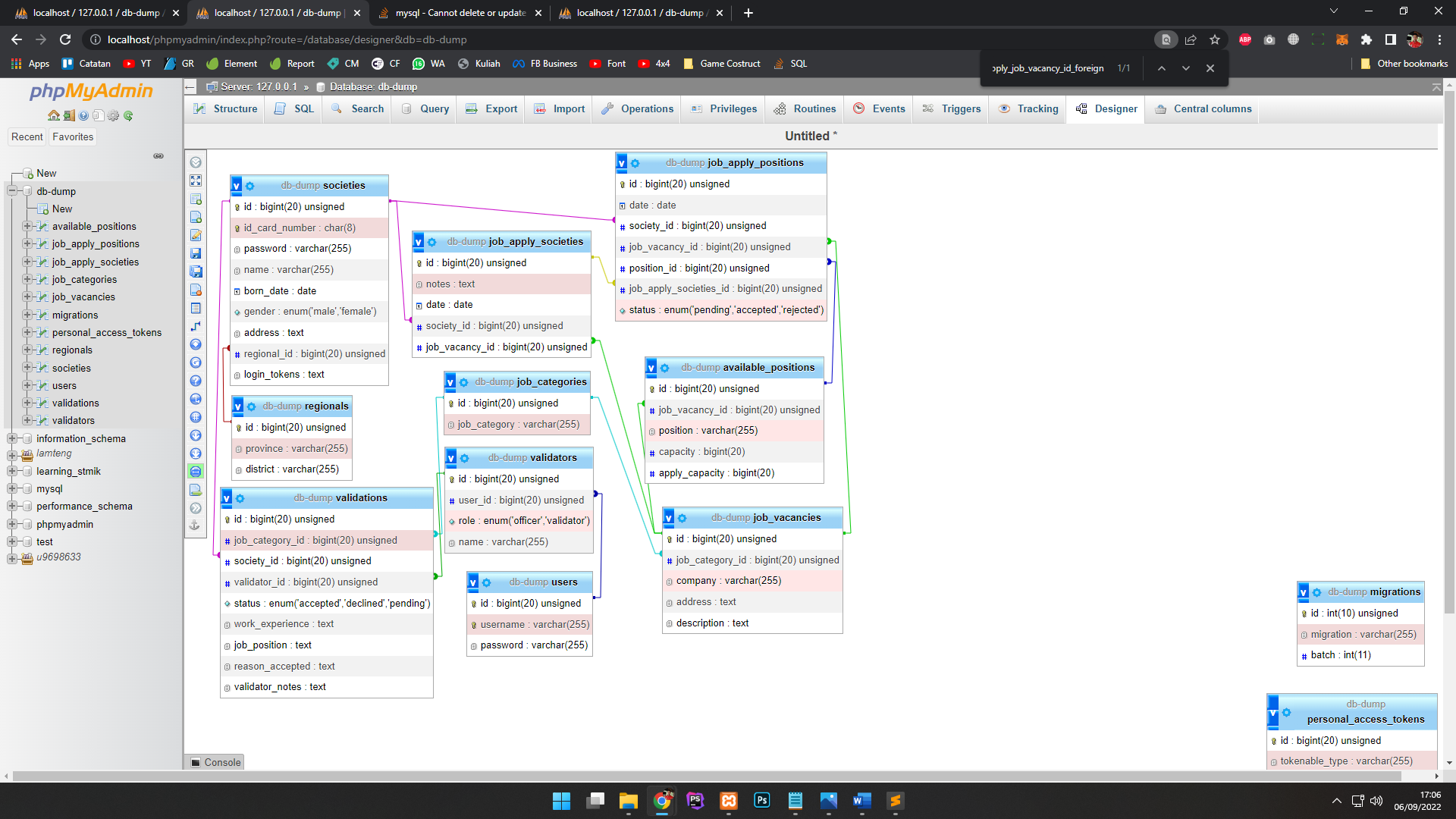
A4g - If invalid token

Header: Response code: 401

Body: { "message": "Unauthorized user" }

# ERD

You can use and improve ERD below:



# INSTRUCTION FOR COMPETITORS

* Import the database that has been provided in db-dump in the media files folder.
* Save your work in the "rest\_api" folder **XX\_module\_rest\_api**

## Postman environment and collection provided in REST\_API\_MEDIA/postman/

## You can check community accounts based on vaccination stages by looking at the accounts that have been provided in the folder REST\_API\_MEDIA/account/

## Use the gui templates provided in the media files.

## Create/generate a DB-diagram named “db-diagram.xxx” (xxx is the extension/type of the file eg. pdf or jpg) and put it into the directory mentioned above. Example:

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