Passerelles numériques
A Gateway for Life

Laravel final exam

EXAM RULE:

- DOCUMENT HAS ALLOWED
- NO CHAT, TALK AND LOOK TO OTHER STUDENTS IF SAW YOU WILL FAILED AUTOMATICALLY
- WRONG SUBMITION DOCUMENT NOT ALLOW TO SUBMIT AGAIN
- REMOVE **Vendors** folder before zip
- EXPORT DATABASE PUT WITH PROJECT
- BEFORE SUBMIT YOU NEED TO RENAME YOUR FOLDER: **FIRST-LASTNAME.ZIP**

PART	SCORE
PART1 - Theory	15
PART 2 - Create Migrations, models, controllers	10
PART 3 - Create the BASIC API	20
PART 4 - Populate your database	12
PART 5 - Create the ADVANCED API	48

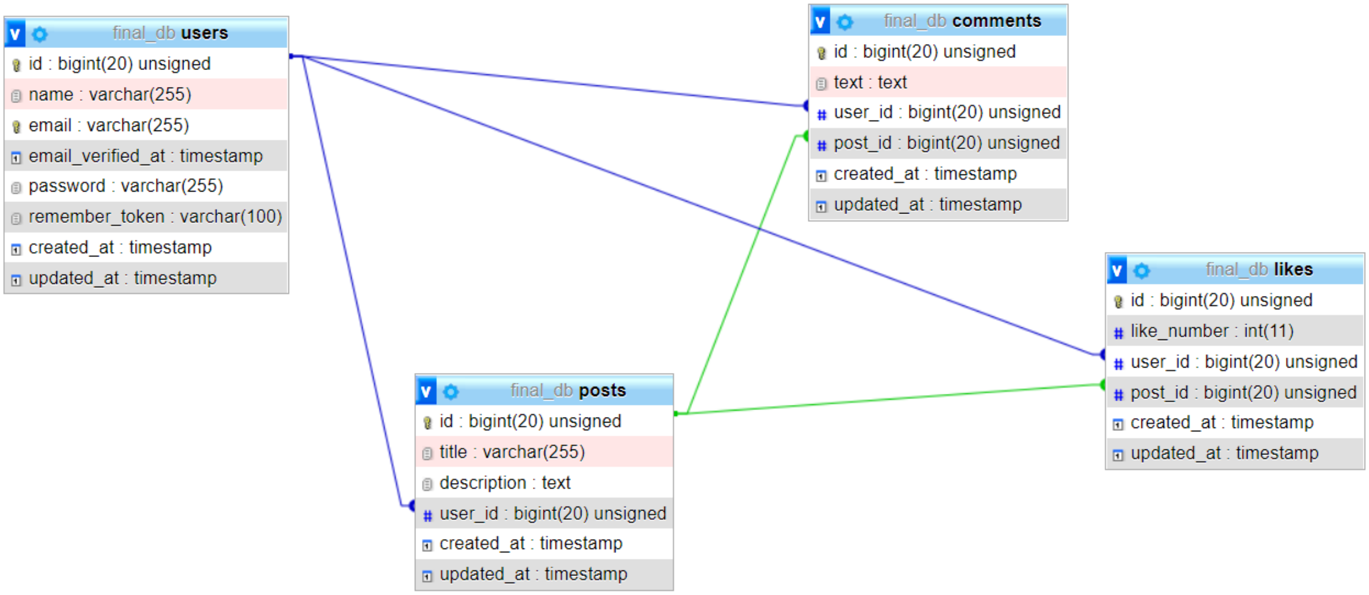
PART 1 – Theory

Google Form: https://forms.gle/p4oDMAx9uVPJjvzM9

PART 2 – Create Migrations, models, controllers

We want to manage **users**, **posts, comments** and **likes** following the bellow ERD.

Create the related migrations, models and controllers on you LARAVEL project and the related database on MySQL.



final_db **users**
- id : bigint(20) unsigned
- name : varchar(255)
- email : varchar(255)
- email_verified_at : timestamp
- password : varchar(255)
- remember_token : varchar(100)
- created_at : timestamp
- updated_at : timestamp

final_db **comments**
- id : bigint(20) unsigned
- text : text
- user_id : bigint(20) unsigned
- post_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

final_db **likes**
- id : bigint(20) unsigned
- like_number : int(11)
- user_id : bigint(20) unsigned
- post_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

final_db **posts**
- id : bigint(20) unsigned
- title : varchar(255)
- description : text
- user_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

PART 3 – Create the BASIC API

Update your project to provide the following API to **manage users, post, and comments**

USERS

HTTP Method	Route	Definition
GET	/users	Get all users
POST	/users	Create new user

POSTS

HTTP Method	**Route**	**Definition**
GET	/posts	Get all posts
POST	/ posts	Create new post

COMMENTS

HTTP Method	**Route**	**Definition**
GET	/comments	Get all comments
POST	/comments	Create new comment

LIKES

HTTP Method	**Route**	**Definition**
GET	/likes	Get all likes
POST	/likes	Create new like

PART 4 – Populate your database

By using this API, create the following element:

USERS

id	Name	Email	password
1	Ronan	ronan@gmail.com	123
2	Rady	rady@gmail.com	234
3	Him	him@gmail.com	234

POSTS

id	user_id	title	description
1	2	Footballer	Ronan is the best defender
2	1	PNC Premier League	Rady is the best striker at PNC league
3	3	Motor	Him is best rider in PNC

COMMENTS

id	post_id	user_id	text
1	1	2	Yeah, great news!
2	2	2	I want to play with you
3	3	1	Well! Done!
4	1	3	I never play with him

LIKES

id	post_id	user_id	like_number
1	1	2	1
2	2	3	1
3	3	1	1
4	1	3	1

PART 5 – Create the ADVANCED API

Update your API to manage the following advanced requests:

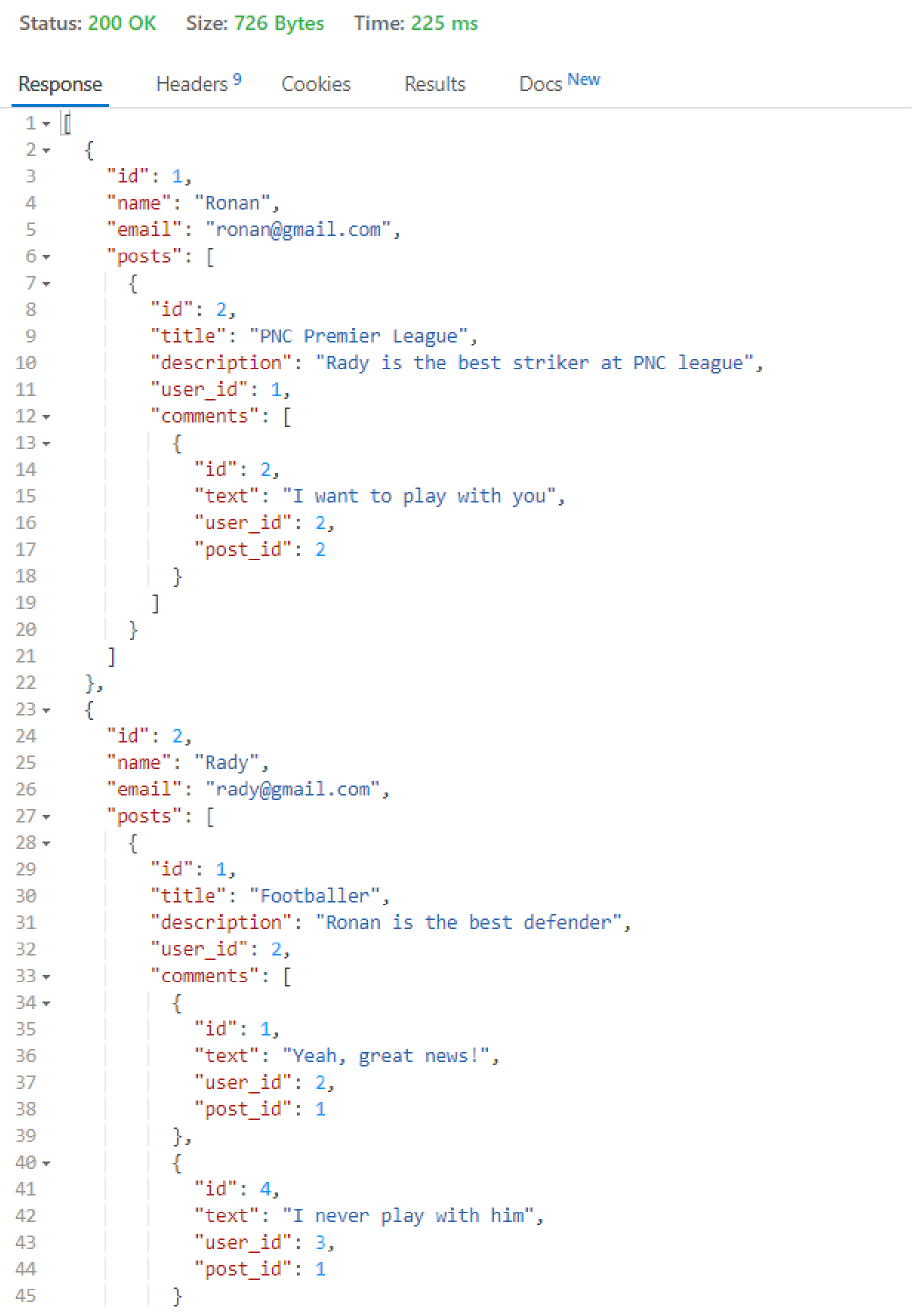
Q1 Users with their post and comments (10 pts.)

HTTP Method	Route	Definition
GET	/users_posts_comments	Get for each users their posts and their comments

Example

Request: **http://127.0.0.1:8000/api/users_post_comments**

Results:



Status: 200 OK   Size: 726 Bytes   Time: 225 ms

```
[
  {
    "id": 1,
    "name": "Ronan",
    "email": "ronan@gmail.com",
    "posts": [
      {
        "id": 2,
        "title": "PNC Premier League",
        "description": "Rady is the best striker at PNC league",
        "user_id": 1,
        "comments": [
          {
            "id": 2,
            "text": "I want to play with you",
            "user_id": 2,
            "post_id": 2
          }
        ]
      }
    ]
  },
  {
    "id": 2,
    "name": "Rady",
    "email": "rady@gmail.com",
    "posts": [
      {
        "id": 1,
        "title": "Footballer",
        "description": "Ronan is the best defender",
        "user_id": 2,
        "comments": [
          {
            "id": 1,
            "text": "Yeah, great news!",
            "user_id": 2,
            "post_id": 1
          },
          {
            "id": 4,
            "text": "I never play with him",
            "user_id": 3,
            "post_id": 1
          }
```

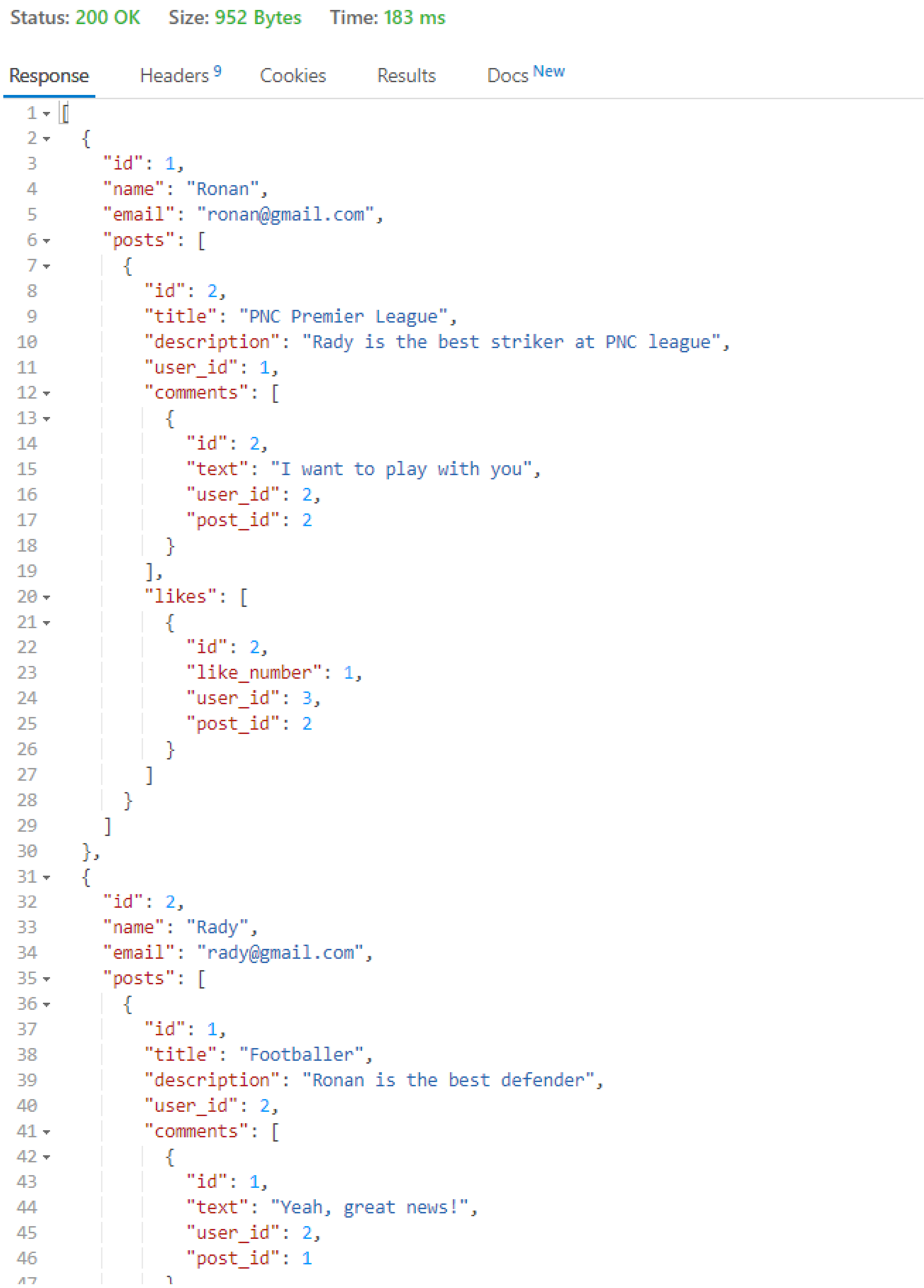
Q2 Users with their post, comments and likes (15 pts.)

HTTP Method	Route	Definition
GET	/users_posts_comments_likes	Get for each users their posts and their comments and likes

Example

Request: **http://127.0.0.1:8000/api/users_posts_comments_likes**

Results:



```
Status: 200 OK    Size: 952 Bytes    Time: 183 ms
```

```
[
  {
    "id": 1,
    "name": "Ronan",
    "email": "ronan@gmail.com",
    "posts": [
      {
        "id": 2,
        "title": "PNC Premier League",
        "description": "Rady is the best striker at PNC league",
        "user_id": 1,
        "comments": [
          {
            "id": 2,
            "text": "I want to play with you",
            "user_id": 2,
            "post_id": 2
          }
        ],
        "likes": [
          {
            "id": 2,
            "like_number": 1,
            "user_id": 3,
            "post_id": 2
          }
        ]
      }
    ]
  },
  {
    "id": 2,
    "name": "Rady",
    "email": "rady@gmail.com",
    "posts": [
      {
        "id": 1,
        "title": "Footballer",
        "description": "Ronan is the best defender",
        "user_id": 2,
        "comments": [
          {
            "id": 1,
            "text": "Yeah, great news!",
            "user_id": 2,
            "post_id": 1
          }
```

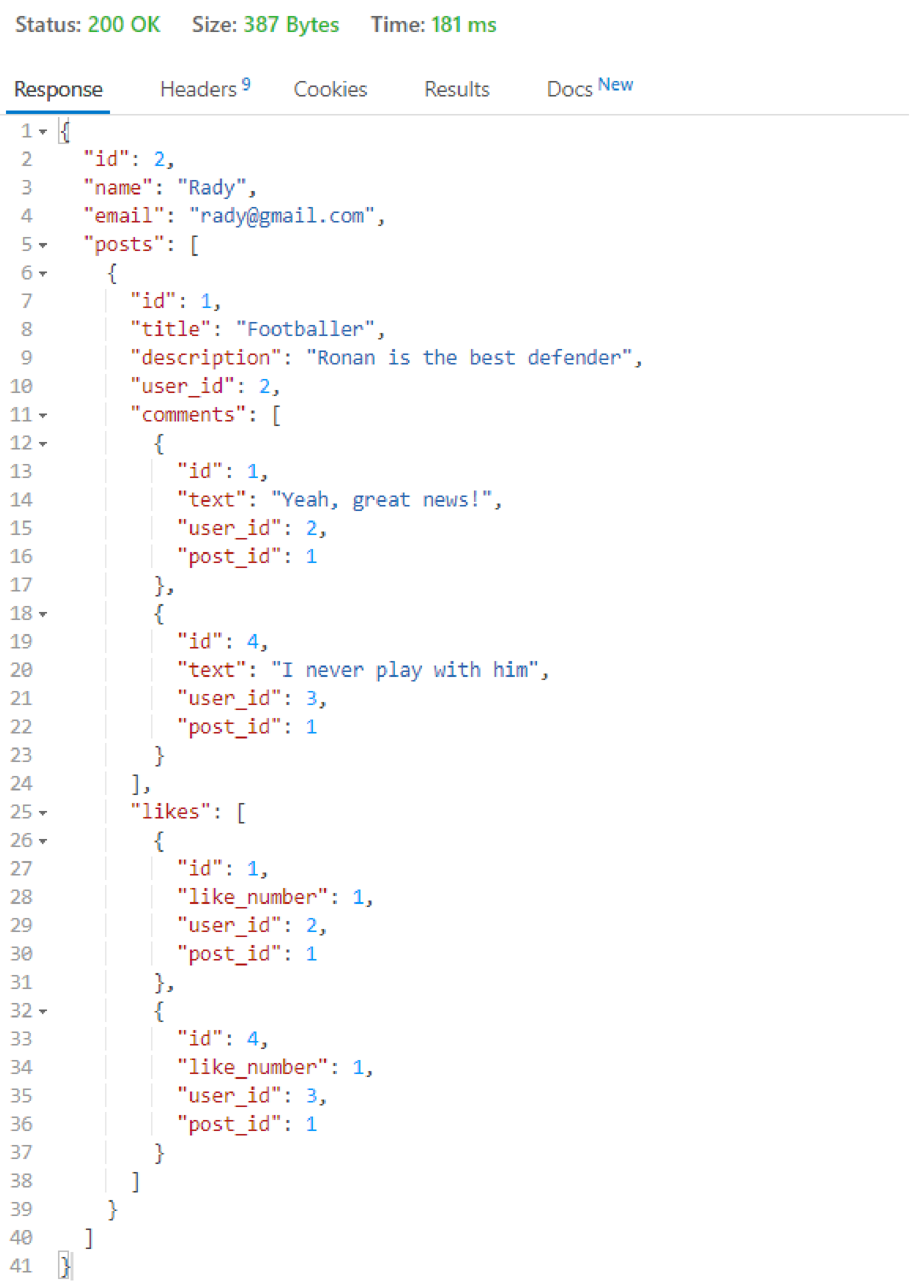
Q3 Single User with their post, comments and likes (10 pts.)

HTTP Method	Route	Definition
GET	/users_posts_comments_likes/{id}	Get single user with their posts and their comments and likes

Example

Request: **http://127.0.0.1:8000/api/users_posts_comments_likes/2**

Results:



Status: 200 OK Size: 387 Bytes Time: 181 ms

```
{
  "id": 2,
  "name": "Rady",
  "email": "rady@gmail.com",
  "posts": [
    {
      "id": 1,
      "title": "Footballer",
      "description": "Ronan is the best defender",
      "user_id": 2,
      "comments": [
        {
          "id": 1,
          "text": "Yeah, great news!",
          "user_id": 2,
          "post_id": 1
        },
        {
          "id": 4,
          "text": "I never play with him",
          "user_id": 3,
          "post_id": 1
        }
      ],
      "likes": [
        {
          "id": 1,
          "like_number": 1,
          "user_id": 2,
          "post_id": 1
        },
        {
          "id": 4,
          "like_number": 1,
          "user_id": 3,
          "post_id": 1
        }
      ]
    }
  ]
}
```

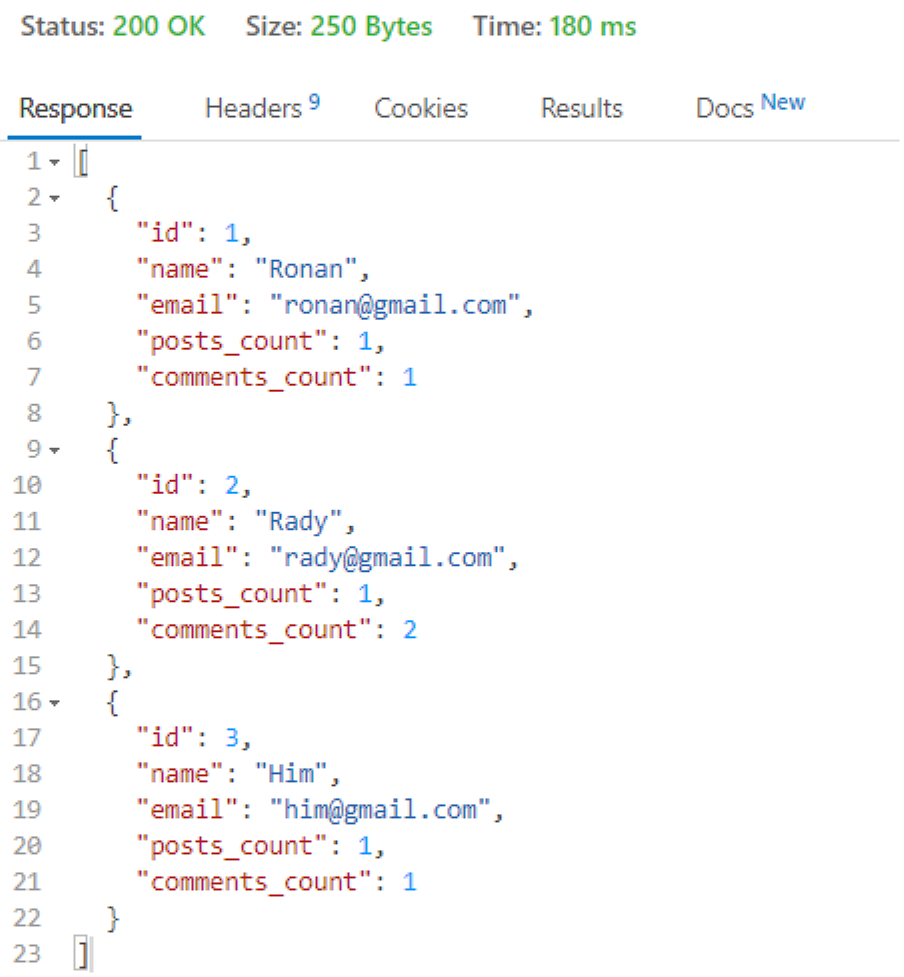
Q4 Count posts and comments of each users (13 pts.)

HTTP Method	Route	Definition
GET	/count_posts_comments	Count posts and comments of each users

Example

Request: **http://127.0.0.1:8000/api/count_posts_comments**

Results:



Status: 200 OK Size: 250 Bytes Time: 180 ms

```
[
  {
    "id": 1,
    "name": "Ronan",
    "email": "ronan@gmail.com",
    "posts_count": 1,
    "comments_count": 1
  },
  {
    "id": 2,
    "name": "Rady",
    "email": "rady@gmail.com",
    "posts_count": 1,
    "comments_count": 2
  },
  {
    "id": 3,
    "name": "Him",
    "email": "him@gmail.com",
    "posts_count": 1,
    "comments_count": 1
  }
]
```