

| **EXAM RULE**:   * CLASSROOM DOCUMENTS INCLUDING SLIDES ARE ALLOWED * NO CHAT, TALK OR LOOK TO OTHER STUDENTS. IF SEEN YOU WILL BE FAILED AUTOMATICALLY * ONLY THE FIRST SUBMISSION WILL BE ACCEPTED IN THE CASE OF MISTAKES * REMOVE **Vendors** folder before zip * EXPORT DATABASE, PUT WITH PROJECT * BEFORE SUBMITTING, 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 |

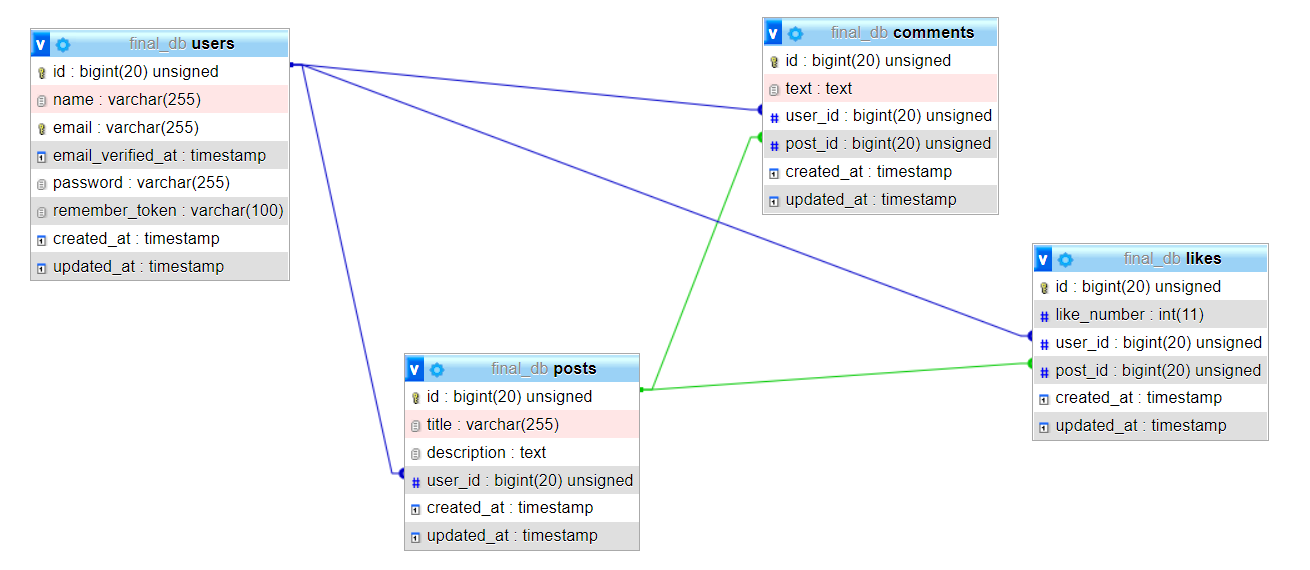
# PART 1 – Theory

Google Form: <https://forms.gle/ttr9sYWBaH51KWV19>

# PART 2 – Create Migrations, models, controllers

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

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



# 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](mailto: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:



**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:



**Q3** SingleUser 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:



**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:

