



ChikChika

Laravel Twitter Clone

[Overview](#)

[Delivery Terms](#)

[Deadline](#)

[Source Code](#)

[Working Project URL](#)

[General Scope](#)

[User Accounts and Profiles](#)

[Registration/Login](#)

[User Profile](#)

[Following Users](#)

[Tweets](#)

[Main Feed](#)

[Tweet Contents](#)

[Replies/Comments](#)

[Likes](#)

[Notifications](#)

[REST API](#)

[Token Generation](#)

[Endpoints](#)

[Additional Scope](#)

Overview

This project aims to develop a Twitter-like social network using a Laravel framework. The focus should be made on the demonstration of your Laravel skills, not on the visual parts of the application, though a nice visual side of the application will be more than welcome.

Important Note: Even though no-SQL databases might be a better fit for your data structure, you should be using a relational database like PostgreSQL or MySQL and take advantage of Eloquent ORM and its' relationships. This is the most important part of your

application - database structuring. Let's see how you will implement the user following and the rest schema...

Delivery Terms

There are several things you should take into account...

Deadline

Since this is a final project that should be assessed the deadline for the project delivery is the last session date of our class. For example, if our final session will be on April 15, then you should send the project by the end of April 15.

Source Code

You should send your project's source code as a ZIP file, with all the necessary components inside it for the installation. After extracting the files from the ZIP archive, I should be able to easily install dependencies via the `composer install` command, run database migrations and seeders and run the project via the `php artisan serve` command.

If your project will lack the dependencies or it wouldn't be possible to run the project as stated above (or the project has tons of errors during the launch) the project will NOT be assessed and it will impact your final score.

Working Project URL

Besides sending the source code, you should also host the project and send me the working URL. It is expected that the project will be accessible on this URL for about 2 weeks from the delivery. This will be the demonstration that you are capable of setting up a project on a server, as well as sharing a demo for your project with the others.

General Scope

You will be building a Twitter clone, where users are registering, posting tweets, and then these tweets can be replied to, liked, added to the list and etc...

User Accounts and Profiles

Registration/Login

Everything begins with the user registering on a website. Please, note... email verification is essential for using your account. A special and unique URL like **`https://chikchika.com/{username}`** should be given to every registered user.

User Profile

The user should be able to decide whether/her profile is private or public. Private profiles are restricted to show tweets only to the users you are following. Public profile tweets can be accessed by everyone, even without authentication.

Following Users

One of the fundamental features is to follow other users. Users can be followed and unfollowed.

Tweets

Main Feed

This is where all the tweets from the users you are following are visible. Tweets should be ordered by their creation date, most recent on top. When you scroll down the page the “Load More” button should be visible and load the next batch of tweets if clicked.

Tweet Contents

The tweet can contain only text content up to 140 characters. In case URLs are referred in the text, they should be replaced with clickable links.

Replies/Comments

Every tweet can be replied to. Replies should be displayed only in one level of nesting, no need to display reply of a reply to a tweet as a nested reply. The quantity of replies has to be displayed below the tweet.

Likes

Tweets can be liked or unliked. The number of likes should be displayed below the tweet.

Notifications

Users should receive notifications for the following actions:

- The user you are following has tweeted.
- The user has followed you.
- The user has unfollowed you.
- The user has replied to or liked your tweet.
- Aggregated information: “For the past 1 week you followed X users, Y users followed you.”

The notifications bar should have a number of unread notifications. Notifications should be marked as read.

REST API

The users' data can be accessed via the REST API. Authentication should be based on API tokens.

Token Generation

A special page should exist to generate and display (not store in the database) an API token. This token will later be used in the API endpoint requests.

Endpoints

- **GET /v1/me** - returns current user details
- **GET /v1/me/following** - returns the list of users you are following to
- **GET /v1/me/follows** - returns the list of users you are followed by
- **GET /v1/tweets** - returns the feed
- **POST /v1/tweets** - creates a tweet
- **GET /v1/tweets/{tweet_id}** - returns the single tweet
- **GET /v1/tweets/{tweet_id}/replies** - returns the replies made on a single tweet
- **POST /v1/tweets/{tweet_id}/like** - likes a tweet
- **DELETE /v1/tweets/{tweet_id}/unlike** - unlikes a tweet
- **POST /v1/tweets/{tweet_id}/reply** - replies to a tweet

Additional Scope

The scope is not final and please be prepared that along with the Laravel materials I will be explaining and teaching your project might need to add additional functionalities or refactor the existing code.

The main purpose of this is that the project scope is almost never fixed in the real world, so you should be ready to modify your code as needed.