## Introduction

The projects are designed to have progressive development, starting with a simple backend to let the student be focused on user interface design (UI), these interfaces can be as complex as the student wants to develop using skills learned in the bootcamp.

Once basic requirements have been completed and user interface has been developed for applications, the student has the option to create new endpoints to add new features to the application.

## Development

#### Stage 1 (base development)

**Backend:**

1. Read endpoint: To get data from database it will be returned on JSON format to the client.

**Frontend**:

1. UI is open design development. Design and implementation depends on the students.
2. Create API calls needed to get all data from the server.

#### Stage 2 (New features)

**Backend**:

Possible endpoints to develop:

1. Endpoint to insert new entries to the DB.
2. Endpoint to update items on the DB.
3. Endpoint to delete an entry on the DB.

**Frontend:**

1. Get every together, use the new endpoints developed and add the new features to the IU.

#### Stage 3 (Optional)

1. Add a search function on your application.
2. Add sort functionality to the lists on your application.

## Project

### Photo Board

Microblogging comes popular on the 2000s with Twitter, Microblogging enables users to express themselves by short messages, not more than 145 chars, the target on this project is to build a simple timeline and on the second version add a feature to give users add a new message.

**Database:**

Table: messages

Fields: id, message, created\_at

**Stage 1:**

*Backend*

Endpoint: GET /messages

Description: Endpoint to get *n* messages from your database.

Suggested response

*HTTP 200*

{

id: int,

message: string,

date: string

}

*FrontEnd*

UI: Develop your own microblogging timeline, be creative using all tools that you learned at the bootcamp.

Data source: Use endpoints developed by you to get data needed.

**Stage 2:**

Feature: Messaging.

Description: Enable to add o remove messages from your timeline.

*Backend*

Endpoint: POST /message

Description: Add a new message.

Suggested response: *HTTP 201*

Endpoint: DELETE /message/:message\_id

Description: Remove a message from the timeline.

Suggested response: *HTTP 204*

*FrontEnd*

UI: Add the new functionality to you UI.