## 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

Create a photo board Instagram-like. For this project you have to focus on a single panel of photos, be creative and build your own board. The data of photos will be read from a database and will be delivered to the client in JSON format.

**Database:**

Table: photos

Fields: id, url, description, created\_at

Table: likes

Fields: id, photo\_id, created\_at, deleted\_at

**Stage 1:**

*Backend*

Endpoint: GET /photos

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

Suggested response

*HTTP 200*

{

id: int,

url: string,

description: string

}

*FrontEnd*

UI: Develop your own photo board, 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: Like

Description: Enable your application to receive likes for your photos, and if your user changes their minds they have the option to remove a like.

*Backend*

Endpoint: POST /like/:photo\_id

Description: Add a like to a photo.

Suggested response: *HTTP 201*

Endpoint: DELETE /like/:photo\_id

Description: Remove a like from a photo, consider a soft delete option.

Suggested response: *HTTP 204*

*FrontEnd*

UI: Add the new functionality to you UI, and keep likes count on each photo.