## 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

### Food Finder

People that love food always look for restaurants that were reviewed before for a lot of people, on this project you have to build a list of restaurants, and on version two add functionality to add and remove reviews.

**Database:**

Table: restaurants

Fields: id, title, description, image\_url, created\_at

Table: reviews

Fields: id, restaurant\_id, review, created\_at, deleted\_at

**Stage 1:**

*Backend*

Endpoint: GET /restaurants

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

Suggested response

*HTTP 200*

{

id: int,

title: string,

description: string,

image\_url: string

}

*FrontEnd*

UI: Develop your own restaurant list, 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: Review

Description: Enable your application to add and remove reviews of restaurants.

*Backend*

Endpoint: POST /review/:restaurant\_id

Description: Add a like to a photo.

Suggested response: *HTTP 201*

Endpoint: DELETE /review/:review\_id

Description: Remove a review of a restaurant, consider a soft delete option.

Suggested response: *HTTP 204*

*FrontEnd*

UI: Add the new functionality to you UI.