**Lab 2 Documentation**

Please use this document to briefly explain how your application meets the requirements of the lab. Your explanation should be concise and may include specific information such as the file name, OOP class, or method used. This document serves as a checklist to ensure that your lab has met all the requirements. It also helps me better understand your lab when I grade it.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Complete?** | **Explanation** |
| 1. Create the basic structure of the API application using the **Slim** |  | The basic structure of our API was implemented with the Slim framework. |
| 1. Set up **Eloquent** in **Slim** |  | Eloquent was set up; eloquent.php file was created and added with bootstrap.php |
| 1. Create endpoints to retrieve data for at least three resources of your project. For each resource, you need to retrieve all records and a specific record. |  | Endpoints were created to handle requests like retrieving recipes, recipe categories, cuisines, and ingredients data with the help of Slim. Some methods include getRecipes, etc. |
| 1. Handle at least one one-to-many relationship. |  | The one-to-many inverse relationship handled in our API include the Recipe/RecipeCategory and Category to RecipeCategory along with the relationship between Cuisine and RecipeCuisine. |
| 1. Handle at least one many-to-many relationship. |  | A many-to-many relationship would include the recipe and ingredients. Many recipes can have many ingredients. |
| 1. Output data in JSON format. |  | Postman collection was used to get data in JSON format to view all categories, view all dietary information, view all recipe cuisines, and view all ingredients |