**Spoonacular-Food-App**

**Technical Documentation**

**Requirements:**

Python==3.8

requests==2.25.1

python-dotenv=0.18.0

**Directories:**

**Spoonacular-food-app**- our main root folder which contains all the program files

**Controller**- Has the main program logic recipes.py and rest\_controller.py which fetches and shows the ingredients to the user

**Screenshots-** Screenshots to show some how to execute commands in the terminal. Used these png files in readme.md file

**Tests**- have the automated test cases and one test\_suite.py file to run all the test cases in single suite

**Utils**- important app constants and json parsing logic functions are present in the utils folder

**Documents-** have technical documentation and sample manual test cases logs

**Spoonacular-food-app**:

**Requirements.txt-** dependencies that need to be installed are present in this file. We will first install all these requirements before running the program

**.env-** This file is not pushed into github. But this has only one line of code. Create this file or just rename env.example to .env file and replace the dummy API key in env.example with your API key. Without the proper key you will get an unauthorized access error. Make sure you do not commit this file in github.

**env.example-** This is a dummy file to show how a .env file should look like.

**.gitignore-** This file contains the extensions which should not be pushed to github.

**Main.py-**  our program starts here and this file calls get\_recipes() function

**Controller:**

**recipes.py-**  Main business logic that interacts with user.

**rest\_controller.py-** To call the get request and retrieve Json response.

**Tests:**

**mock\_data.py-**  It has sample test mock data to test the functions.

**test\_json\_parser.py-** It has test cases to test the json\_parser.py functionality. To test these cases separately, Navigate to tests folder and run python3 test\_json\_parser.py

**test\_recipes.py**- It has test cases to test the recipes.py functionality. To test these cases separately, Navigate to tests folder and run python3 test\_recipes.py

**test\_rest\_controller.py**- It has test cases to test the recipes.py functionality. To test these cases separately, Navigate to tests folder and run python3 test\_rest\_controller.py

**Test\_suite.py**- to run all the 3 test case files Navigate to tests folder and run python3 test\_rest\_controller.py navigate to tests folder and run python3 test\_suit.py

**Utils:**

**app\_constants.py-** used to keep track of all the constants

**json\_parser.py-** has functions to parse the recipe json response and store the used, missing and title objects separately.