**Team Members:** Bryan, Cohen, Emily, and Jason

**Project title:** Recipe and Wine Pairer

**Project Description:** Application that allows users to find a recipe based selected ingredients and pairs the recipe with a wine.

**User Story:**

As a person who never has a stocked refrigerator, I want to enter random ingredients to receive a recipe based on those ingredients and a potential wine pairing, with option to see wine scores for each vintage year of the selected wine.

**APIs:**

<https://spoonacular.com/food-api/docs#Get-Wine-Pairing>

<https://globalwinescore.docs.apiary.io/#introduction/format>

**Project Guidelines:**

1. Must use at least two server-side APIs
2. Must use a CSS framework OTHER than Bootstrap
3. Use at least a new third-party API
4. Must have a polished, mobile-first UI
5. Must meet quality coding standards (indentation, scoping, naming, etc.)
6. Does NOT use alerts, confirms, or prompts (look into modals)
7. Must be deployed to GitHub Pages
8. Must be interactive (accept and respond to user input)

# Document version

|  |  |  |
| --- | --- | --- |
| **Version** | **Changed By** | **Description** |
| 1.0 | Jason Miller | Initial draft |
|  |  |  |

# Suggested Approach

**Spoonacular** allows a user to input an ingredients list and will return both recipes and wine recommendations.

**Wine Scores** allows a user to input a wine ID (from spoonacular return) and receive a list of vintage years of that wine with that year’s wine score.

The user inputs up to five ingredients.

The five ingredients are used in the spoonacular call.

Spoonacular then returns a list of objects which are together individually as:

* **Recipe title**
* **Recipe link**
* **Wine titles (array)**

To the user, recipes render as a list of clickable titles, and the top three wine recommendation display as links as well.

When the user clicks a recipe link, she is taken to the recipe website.

When the user clicks a wine link, she can see the wine scores of each vintage of the wine.

The options should be held in local storage under the next ingredient list is entered, which clears local storage and inserts new {type/title/link} objects for both lists.

Based on that approach, we would need the following core elements:

|  |  |  |
| --- | --- | --- |
| **Element/Function** | **Name** | **Description** |
| Array | ingredList | Holds ingredients input by user |
| Input | ingredInput | Allows user to input one ingredient at a time (up to five) |
| Para | ingredDisplay | Displays the list of added ingredients to user |
| Button | ingredAddBut | Allows user to add ingred entered into ingredInput and show in ingredDisplay |
| Button | ingredClearBut | Clears all ingreds from inputDisplay |
| Button | ingredSubmit | Launches both APIs and clears the ingredDisplay |
| Array | foodWineList | Holds objects from spoonacular return |
| Div | ID = recipeReturn | Empty div that is auto-populated upon API recipe return |
| Div | ID = vendorReturn | Empty div that is auto-populated upon API vendor location return |
| Listener | For ingredAdd | Upon click for add button |
| Listener | For ingredClear | Upon click for clear button |
| Listener | For ingredSubmit | Upon click for submit button |
| Listener | For recipe links | Upon click of returned recipe title/link |
| Listener | For wine links | Upon click of returned wine name |
| Function | renderPage | Builds initial screen for first-time user and for return user (pulling from local storage) |
| Function | ingredAdd | Allows user to add ingred entered into ingredInput and show in ingredDisplay |
| Function | ingredClear | Clears all ingreds from inputDisplay |
| Function | ingredSubmit | Launches both spoonacular APIs and displays results |
| Function | recipeLink | Opens web page with full recipe |
| Function | vendorLink | Launches wine API and displays vender/location results |

Based on this description and these fields, we would need to assign the following tasks:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Type** | **Assign** | **Notes** |
| Finalized requires | All |  |  |
| Create mobile layout wireframe | Front end |  | Links not images? |
| Create responsive layout wireframe | Front end |  | Images not links? |
| Select color palette | Front end |  |  |
| Select CSS framework | Front end |  |  |
| Build additional/custom CSS entries | Front end |  |  |
| Build html interface | Front end |  |  |
| Create API call to spoonacular | Back end |  | Includes local storage of returns; returns 5 recipes |
| Create API call to wine-searcher | Back end |  |  |
| Create function - renderPage | Back end |  |  |
| Create function/listener - ingredAdd | Back end |  |  |
| Create function/listener - ingredClear | Back end |  |  |
| Create function/listener - ingredSubmit | Back end |  | Includes two API calls to spoonacular |
| Create function/listener – recipe link | Back end |  |  |
| Create function/listener – wine link | Back end |  |  |
| Unit testing – front end | Front end |  |  |
| Unit testing – back end | Back end |  |  |
| Full testing – interface | Back end |  |  |
| Full testing – functionality | Front end |  |  |
| Device testing - mobile | TBD |  |  |
| Device testing - tablet | TBD |  |  |
| Device testing – laptop | TBD |  |  |
| Device testing - desktop | TBD |  |  |
| Create presentation deck | TBD |  |  |
| Practice presentation | All |  |  |

# appendix a: DOCUMENTATION

<https://spoonacular.com/food-api/docs#Search-Recipes-by-Ingredients>

<https://spoonacular.com/food-api/docs#Get-Wine-Pairing>

Account Key ([jaericm@yahoo.com](mailto:jaericm@yahoo.com) / bootcamp): acc2d918d19f494f9490b92a1b73fc4d

**Sample recipe call and return:**

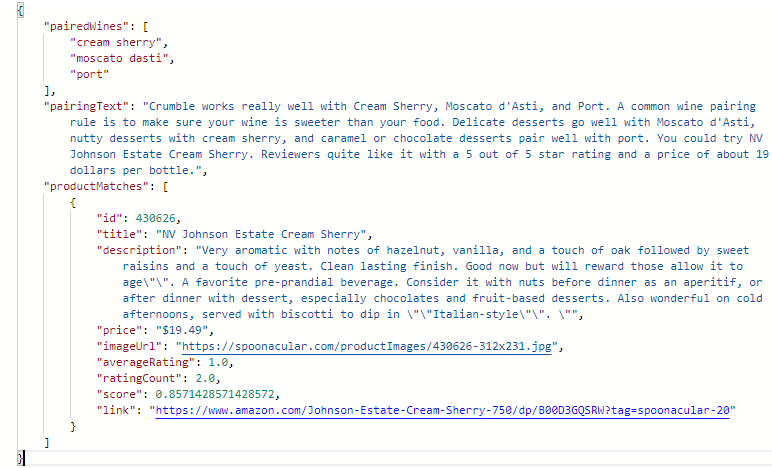
<https://api.spoonacular.com/recipes/findByIngredients?ingredients=apples,+flour,+sugar&number=1&apiKey=acc2d918d19f494f9490b92a1b73fc4d>



**Sample wine recommendation call and return:**

<https://api.spoonacular.com/food/wine/pairing?food=Crumble&apiKey=acc2d918d19f494f9490b92a1b73fc4d>

NOTE: maximum price could also be entered as an optional variable



**Sample wine score call and return:**

<https://www.globalwinescore.com/account/api/> APIkey ([jaericm@yahoo.com](mailto:jaericm@yahoo.com) / bootcamp): Token 02fe26dc4dad3cb9877531fe9ff4b59b0134be8e

<https://api.globalwinescore.com/globalwinescores/latest/?wine_id=50217>

Header

Authorization:Token 02fe26dc4dad3cb9877531fe9ff4b59b0134be8e

Accept:application/json

