## Term project – Undergrads -- A food bookmarking and recommendation application

1) Preface
- The term project for SE481 is designed to encapsulate the IR process insights explored throughout the course and to create an engaging web application.

2) Setup
- Individual project work
- Use the data available at https://www.kaggle.com/datasets/irkaal/foodcom-recipes-and-reviews/data
- You will make an onsite demonstration for evaluation.
- Assigned date: February 19th, 2024.
- Code submission due: March 17th, 2024 (approximately 4 weeks duration)
- (Tentative) Demonstration date: March 18th, 2024, between 9.00 to 12.00

3) Use cases

### UC-001: User authentication
- Description: The user shall have the ability to securely log in to and log out of the food bookmarking and recommendation application.
- Precondition: The user must be registered with the application.
- Postcondition: User is authenticated and has access to personalized features.

### UC-002: Recipe search functionality
- Description: The user shall be able to search for food recipes by **name**, **ingredient**, or **cooking process** in English. The application will **allow for typographical errors** in the search query and will provide results ranked by similarity to the search terms.
- Precondition: User is logged in to the application.
- Postcondition: Search results are displayed to the user in ranked order based on similarity.

### UC-003: Display search results
- Description: The application shall display search results to the user, showing the returned products with both images and names in a card format.
- Precondition: A search query has been executed.
- Postcondition: Search results are visible to the user for further interaction.

### UC-004: Detailed dish information
- Description: The user shall be able to select a dish from the search results to view detailed information including the dish's name, recipe, and cooking steps in a modal dialogue.
- Precondition: Search results are displayed.
- Postcondition: Detailed information about the selected dish is displayed.

### UC-005: Folder management

- Description: The user shall be able to create, manage, and organize multiple personal folders (categories) within the application for bookmarking purposes.
- Precondition: User is logged in.
- Postcondition: New folders are created and visible to the user.

## UC-006: Bookmarking and rating
- Description: The user shall have the functionality to bookmark a dish into a specific folder and provide a rating for the dish ranging from 1 (lowest) to 5 (highest) stars. Also the user shall be able to see their bookmark page where all the folders are shown at once.
- Precondition: The dish is selected from search results.
- Postcondition: The dish is bookmarked in the chosen folder with a user-assigned rating.

## UC-007: Personalized recommendations
- Description: At the landing page, the user shall see their personalized recommendation dynamically generated and displayed a list of recommended dishes to the user, which varies as per the folder selected. The recommendations shall include a summary from all folders, a random selection from a chosen category, and a set of completely random dishes.
- Precondition: User has created folders and bookmarked dishes.
- Postcondition: A personalized list of recommended dishes is displayed on the user's landing page.

## UC-008: Suggestion List Generation
- Description: Upon user request for suggestions, the application shall generate a new list of recommended dishes. This list will be ranked by a learn-to-rank model.
- Precondition: User requests suggestions by viewing a created folder and the folder is not empty.
- Postcondition: A new, ranked list of recommendations is displayed to the user.

4) Scoring rubrics (of 20 points)

| Item | To earn 1 point | To earn 2 points |
|---|---|---|
| It is a web application | The landing page displays three lists of dishes as specified in the use case description. | (1) plus, authentication functionality. |
| Search | The user can search by dish name, ingredient, or cooking process individually. Note all must be implemented. | The user can search by any combination of dish name, ingredient, and cooking process simultaneously. |
| Spell collection | Spell collection is automatically launched once a typo is identified. | A well-designed UX that prompts the user to confirm if the suggested corrections for typos should be used. |
| Detailed dish information | A modal containing complete dish information as described in the use case is provided, along with necessary UX elements. | - |
| Folder management | Folder management functionality is complete as described in the use case and includes necessary UX elements. | - |
| Bookmarking | The user can bookmark a dish into their specific folder. | The user can assign a rating along with their bookmarking. |

| Item | To earn 1 point | To earn 2 points |
| --- | --- | --- |
| Bookmark viewing | The user can view their bookmarked dishes from all folders on one page. | All folders are correctly displayed on one page and bookmarks are ranked, e.g., by the user's average rating |
| Suggestion on landing page | Suggestions are correctly displayed on the landing page. | - |
| More advanced suggestion | - | There is a proof that a machine learning approach excluding kNN can be used to demonstrated enhanced results. |
| Configuration Management | Git is used properly following an appropriate workflow. | - |
| Testing | Only unit tests are appropriately applied. | Tests at higher levels than unit testing are conducted and applied. |
| Exciting IR features. | One additional beneficial IR-related feature is implemented and proven useful. | Two additional beneficial IR-related feature is implemented and proven useful. |