PIEMONTE
FONDI STRUTTURALI E DI INVESTIMENTO EUROPEI 2014/2020

fondo sociale europeo

ITS
TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Market project

Programmazione web - Javascript

Docente: Shadi Lahham

in collaborazione con:

ISTITUTI TECNICI SUPERIORI

Ministero dell'Istruzione

UNIONE EUROPEA

REGIONE PIEMONTE

per una crescita intelligente, sostenibile ed inclusiva
www.regione.piemonte.it/europa2020
INIZIATIVA CO-FINANZIATA CON FSE

Market

A weekly product dashboard

Aim

Create a website that displays a weekly list of supermarket goods filtered by expiry date.
This is new project that uses parts of the previous project **'Expiry List'**

Requirements

Structure and behavior

The requirements are similar to the previous project with the following changes:

- the page should have a complete HTML structure and CSS styling
- comments are required in the HTML and CSS files as in Appendix 01
- free to design the site but it should contain the minimum functionality required
- the HTML and CSS structure and style will be evaluated as part of the project
- each week view should have navigation buttons, 'next' and 'previous'
  - they change the view to the next or previous week
  - navigation buttons should be hidden or disabled when not required
- each item's status should have a unique visual style

Settings

- the website should have a settings panel
- the panel should be animated via CSS or Javascript
- all your specific configuration options should be available in the settings panel
  - the attached images are just an example, include your own settings
- the settings panel should have a 'save' button
- when the user saves the settings
  - the panel should close
  - the page information should be updated to reflect the changed settings
- numeric inputs in the settings panel should only accept numbers
- it should be possible to increment numbers from the input UI
- date inputs should be checked for validity
  - define acceptable date formats and check with regular expressions
  - if a date is not valid an error message should be shown below the input
  - alternatively the 'save' button can be disabled if any input is not valid

Important

- variables should have meaningful names
- the code should be well documented
- the code should be well indented and well formatted
- the data structures should be effective and the code should be efficient
- you need to be able to explain each line of your code
- your page should behave similarly to the screenshots provided

Bonus (extra points)

Bonus 1:

- sort each column of the list when the user clicks on the column title

Bonus 2:

- items can have an additional state 'removed' which means that the item has been manually removed by the supermarket manager
    - clicking on the status of an item changes it to 'removed'
    - clicking again changes it back to the original status
    - removed items should be remembered for future weeks

Help

- You can use pieces of code from the previous project, but you will probably need to rewrite them to adapt to the new requirements
- plan your data structures and functions before starting
- use arrays, objects and any other structures you think are necessary
- use a config object to store all settings

Compatibility

The project should be tested and work properly on:

1. **Chrome**
2. **Firefox**
3. **Edge**
4. Compatibility with other browsers is a nice **bonus**

Libraries, frameworks and language features

- do not use any external libraries or frameworks
- you have to write all the code yourself
- if you use any language features not seen in class, they have to be justifiable, correct and tested

Documentation and validation

Comments and code documentation

- all HTML, CSS and JS files should contain comments and be well documented
- HTML files should contain comments to indicate important sections
- CSS files should have a header and contain comments where needed
- JSDoc header documentation for every file
- JSDoc documentation for every function
- follow all the comments and documentation requirements in Appendix 01

Validation

- HTML files should be validated https://validator.w3.org/
- CSS files, if used, should be validated https://jigsaw.w3.org/css-validator/

JSDoc documentation (required)

- Generate a JSDoc documentation for your code and put it in a folder called /JSDoc

Readme

Include a readme.md file that includes at least the following sections

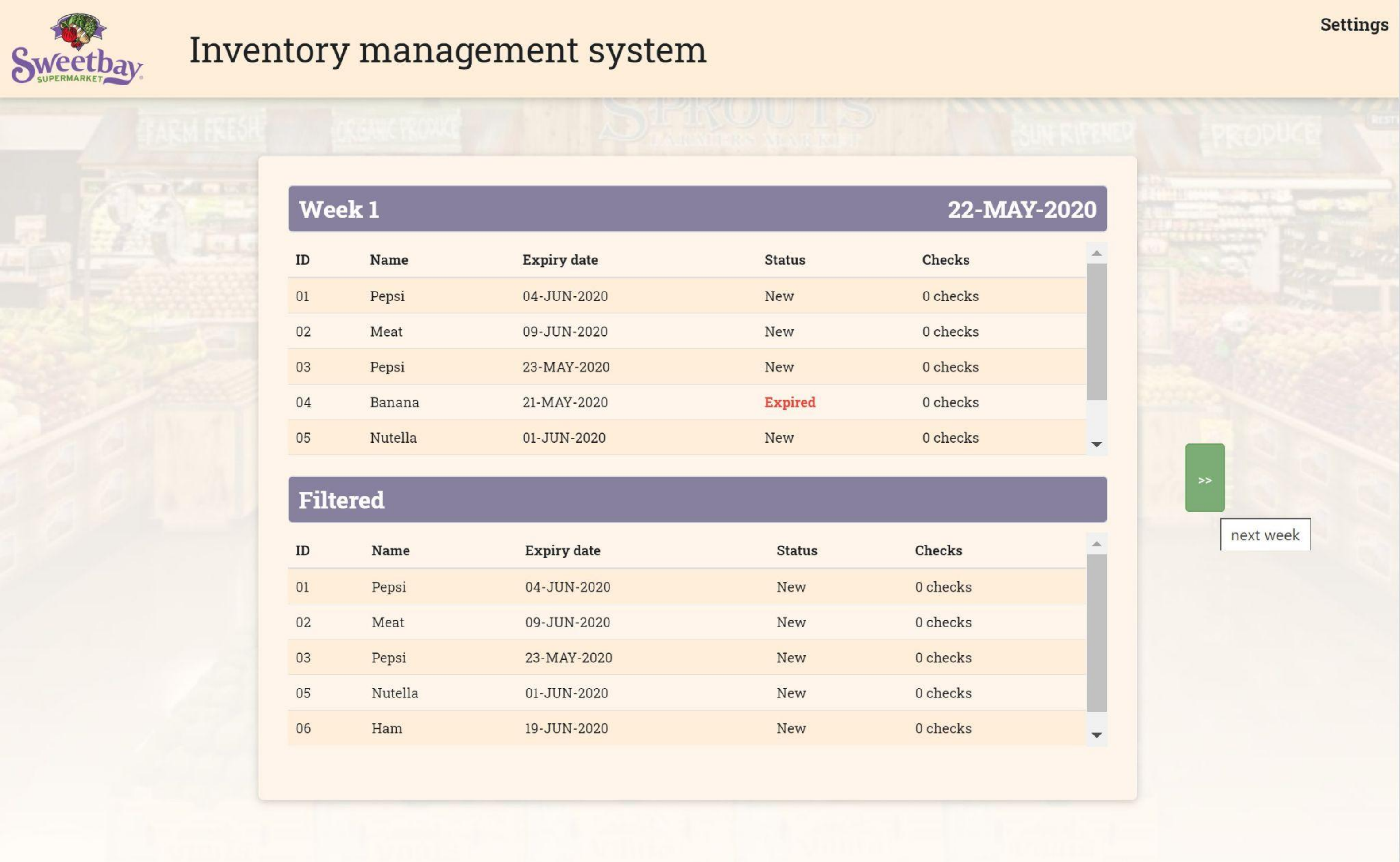
- Introduction / Project description
- Usage (how to set up, run and use the application)
- Configuration and technical characteristics
- Files and project structure
- Features delivered
    - feature 1: description of feature 1
    - feature 2: description of feature 2
    - feature 3: missing
- Bonuses delivered
    - Bonus 1: description of bonus 1
    - Bonus 2: missing
- Browser compatibility
    - Chrome v##.##.##: tested and fully compatible
    - Opera v##.##.##: tested and partially compatible - feature x not working
    - Edge v##.##.##: not tested or not functional
    - The above are only examples, your application should be fully compatible as per the compatibility requirements
- External resources
    - Links and description of external resources
    - e.g. JSON files, APIs, DBs, etc
- License and contact information
- Authors: names, roles and team composition
- Changelog and version history
- Any other information that you think is important

note: "feature" means requirements or behaviors of this project as you have interpreted them from this document.

Screenshots

- the attached screenshots are for reference only
- develop your own style, look and feel
- your application should behave similarly to the screenshots provided

Screenshot 1



Settings

Inventory management system

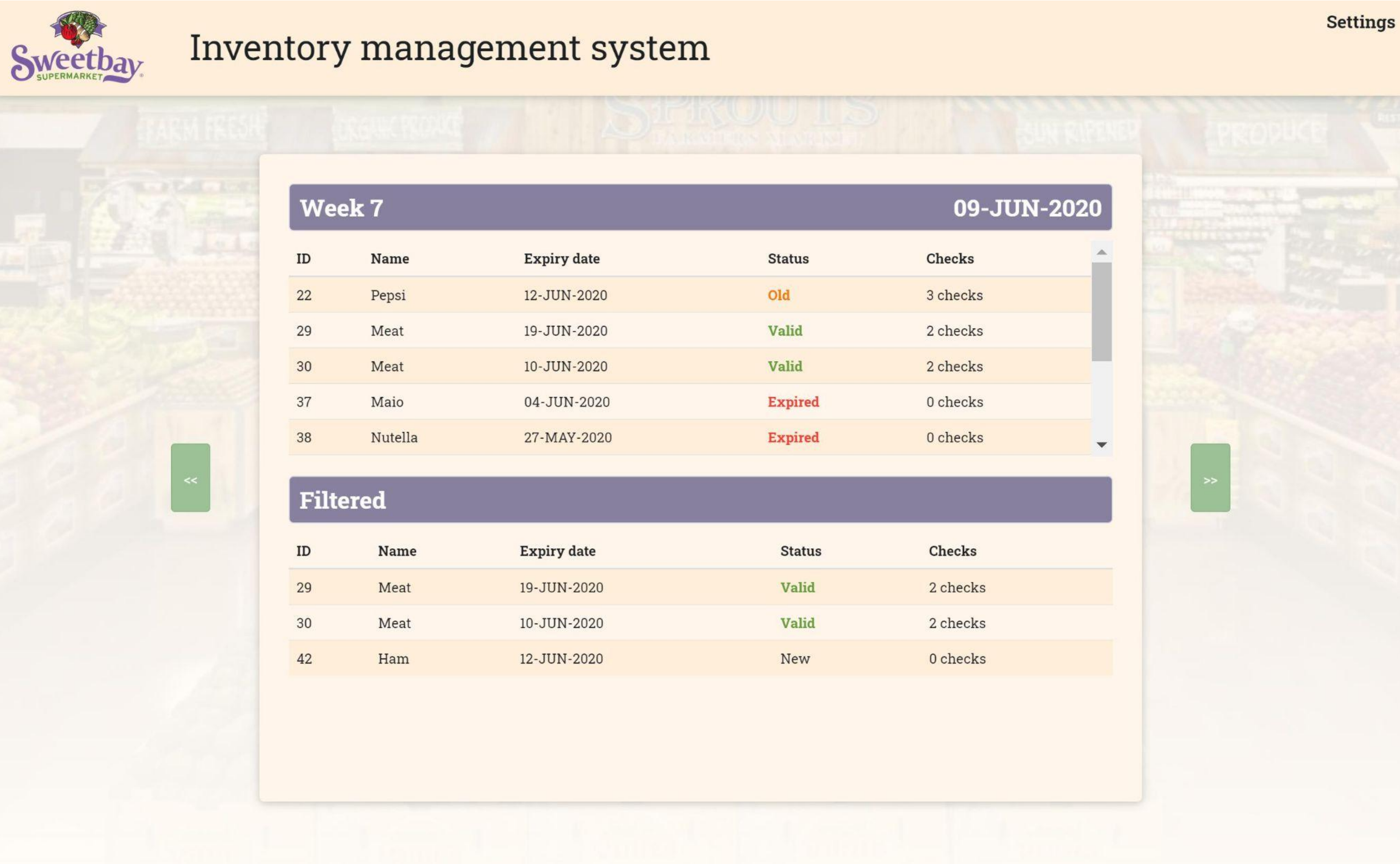
Week 1 — **22-MAY-2020**

ID	Name	Expiry date	Status	Checks
01	Pepsi	04-JUN-2020	New	0 checks
02	Meat	09-JUN-2020	New	0 checks
03	Pepsi	23-MAY-2020	New	0 checks
04	Banana	21-MAY-2020	Expired	0 checks
05	Nutella	01-JUN-2020	New	0 checks

Filtered

ID	Name	Expiry date	Status	Checks
01	Pepsi	04-JUN-2020	New	0 checks
02	Meat	09-JUN-2020	New	0 checks
03	Pepsi	23-MAY-2020	New	0 checks
05	Nutella	01-JUN-2020	New	0 checks
06	Ham	19-JUN-2020	New	0 checks

Screenshot 2



Inventory management system

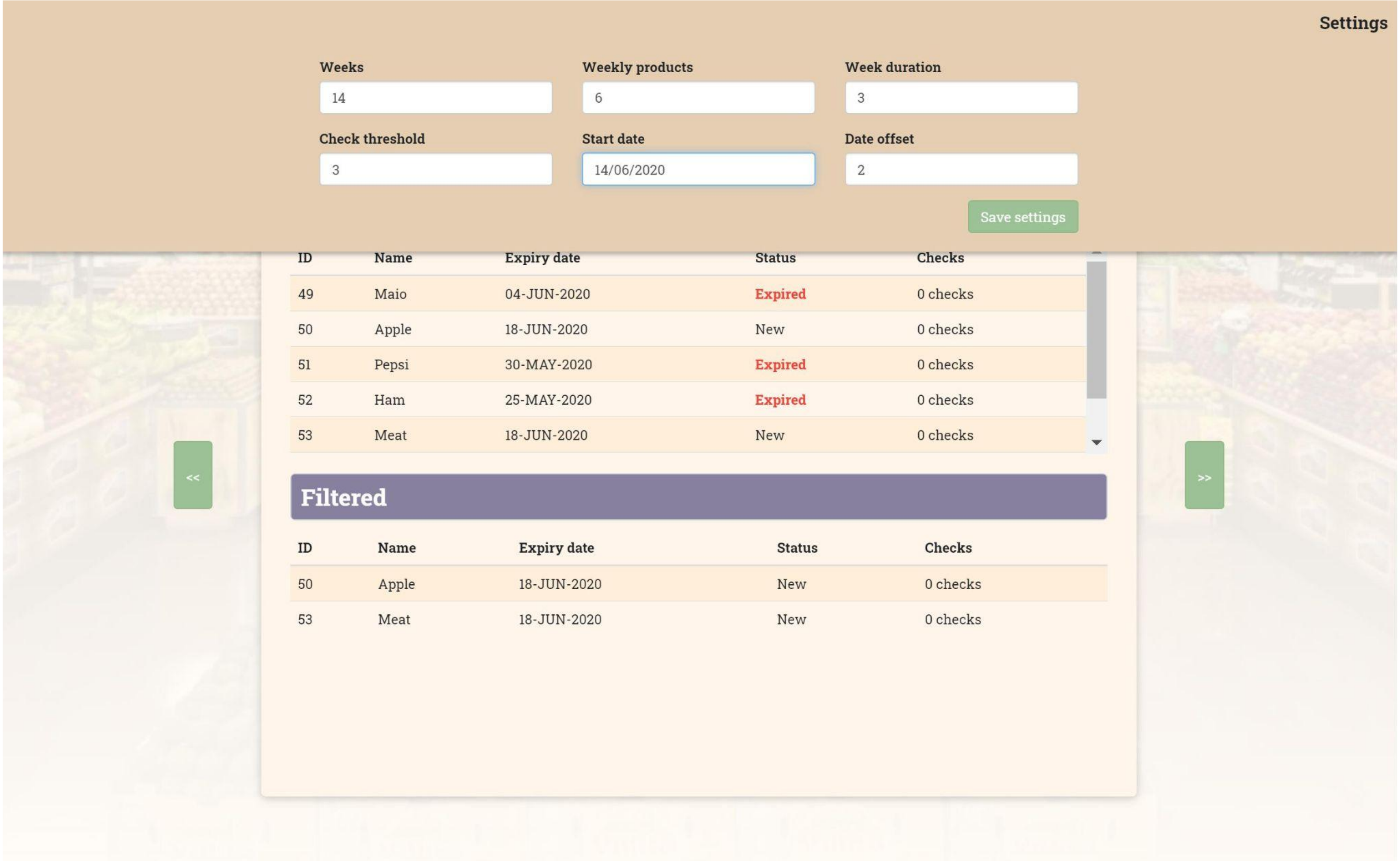
Week 7 09-JUN-2020

ID	Name	Expiry date	Status	Checks
22	Pepsi	12-JUN-2020	Old	3 checks
29	Meat	19-JUN-2020	Valid	2 checks
30	Meat	10-JUN-2020	Valid	2 checks
37	Maio	04-JUN-2020	Expired	0 checks
38	Nutella	27-MAY-2020	Expired	0 checks

Filtered

ID	Name	Expiry date	Status	Checks
29	Meat	19-JUN-2020	Valid	2 checks
30	Meat	10-JUN-2020	Valid	2 checks
42	Ham	12-JUN-2020	New	0 checks

Screenshot 3



Weeks	Weekly products	Week duration
14	6	3
Check threshold	Start date	Date offset
3	14/06/2020	2

ID	Name	Expiry date	Status	Checks
49	Maio	04-JUN-2020	Expired	0 checks
50	Apple	18-JUN-2020	New	0 checks
51	Pepsi	30-MAY-2020	Expired	0 checks
52	Ham	25-MAY-2020	Expired	0 checks
53	Meat	18-JUN-2020	New	0 checks

Filtered

ID	Name	Expiry date	Status	Checks
50	Apple	18-JUN-2020	New	0 checks
53	Meat	18-JUN-2020	New	0 checks