Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Department of Software Engineering and Automation

Web Programming

Laboratory work #1

Learn CSS and HTML

Author:
Valeria Dubina
std. gr. FAF-203

Supervisor:
Alexei Șerșun

Chișinău 2023

Figures

Listings

1 Task

- Download the design as PSD file from Freepik;
- Extract images as JPG files using Aspose or similiar tools.
- Copy index.html, reset.css and style.css to your repo;
- Update index.html and style.css, so that the page looks similar to the design from PSD file.

2 Results

The following Grocery delivery service web template was proposed as a laboratory work:

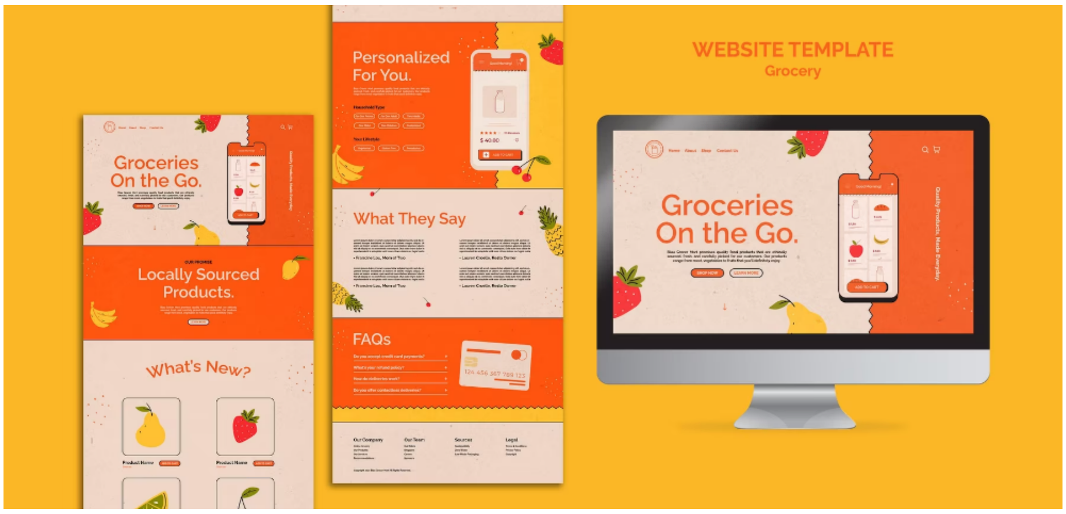


Figure 1: Grocery delivery service web template

After researching the best option for PSD extractors, Figpea [1] was chosen because is simple and straight forward.

To write more simpler and maintainable stylesheets, SCSS was chosen instead of CSS. SCSS (Sass) is a preprocessor scripting language that adds advanced features to CSS, making it more efficient and convenient to write and maintain stylesheets. Some benefits of using SCSS that were implemented in this project are:

- Variables: SCSS allows defining variables to store reusable values, such as colors, fonts, and sizes, making it easier to update styles globally.
- Nesting: SCSS allows nesting CSS rules within one another, making it more intuitive to write and read styles for complex HTML structures.

For example, a color palette was defined as a set of variables, which were reused throughout the page. This leads to less confusion when styling the page and overall to a cleaner stylesheet.

```
        $--orange: #d45b30;
        $--bright-orange: #f4581f;
        $--beige: #f4d5ae;
        $--yellow: #f5bf43;
        $--black: #000000;
```

Listing 1: SCSS color variables

WebStorm IDEA has a built-in SCSS compiler that can automatically compile SCSS files to CSS. To use this feature is important to enable the File Watchers plugin and configure the SCSS file watcher. Once configured, WebStorm IDEA will watch SCSS files for changes and automatically compile them to CSS when you save them, resulting in three files:

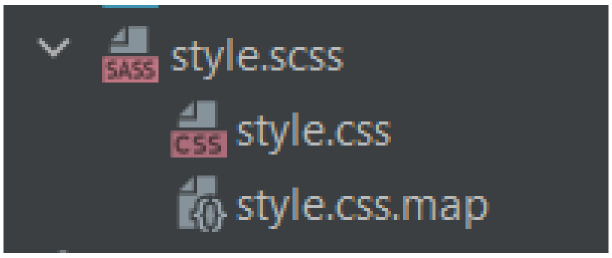


Figure 2: SCSS mapping to CSS in JetBrains

To have a clean working directory, the styles were split into small files, grouped in directories, and imported in style.scss as follows:

```
@import "styles/general/general";
@import "styles/general/text";
@import "styles/general/buttons";
@import "styles/general/card";
@import "styles/general/header";

@import "styles/blocks/product-description";
@import "styles/blocks/promise-page";
@import "styles/blocks/products";
@import "styles/blocks/made-simple";
@import "styles/blocks/faqs";
```

Listing 2: style.scss

The hardest part was figuring out how to set the background color, the little particles of dust, and the content itself. As a solution was chosen to implement three layers, the main class with position relative that contains both the content and texture (with position absolute and z-index of 1), see the listing below.

```
.main {
  /// ...
  position: relative;
  overflow: hidden;
  background: $--beige;

  .texture {
    opacity: 0.6;
    position: absolute;
    left: 0;
    top: 0;
    width: 100%;
    z-index: 1;
  }
  .content {
    position: relative;
  }
```

Listing 3: General CSS layout

Looks great... in theory, but now all the design fruits are also covered in dust but they are not supposed to be. So it is important not to forget to set a z-index bigger than the textures one so they are rendered at the top, here is an example of how this situation was overcome:

```
        .fruit {
          position: absolute;
          z-index: 1;
        }
        .strawberry {
          &--top {
            width: calc(7.5rem - 1vw);
            transform: rotate(-160deg);
            top: -70px;
            right: 32rem;
          }
          &--left-bottom {
            width: calc(10rem - 1vw);
            left: -145px;
            top: calc(100vh - 18rem);
          }
        }
```

Listing 4: SCSS fruits positioning

For element displacement the CSS flexbox and grid. Both Flexbox and Grid offer a wide range of layout and alignment properties that allow creating a layout with just a few lines of code.

The resulting web page consists of several sections, including ”Groceries. On the go”, ”Locally sourced block”, ”What’s new”, ”Made simple”, and ”FAQ”. In condition was not specified to implement the ”What’s new” section, but I wanted to try to use the CSS Grid and this looked like a good section to practice it. For the whole result see ANNEX A, and check Githubfor implementation code.

3 Conclusion

The objective of the laboratory was to develop a web page that resembles the original PSD design for the Grocery delivery service. For that, the process included downloading the PSD file, extracting the necessary assets, and coding the HTML and CSS files.

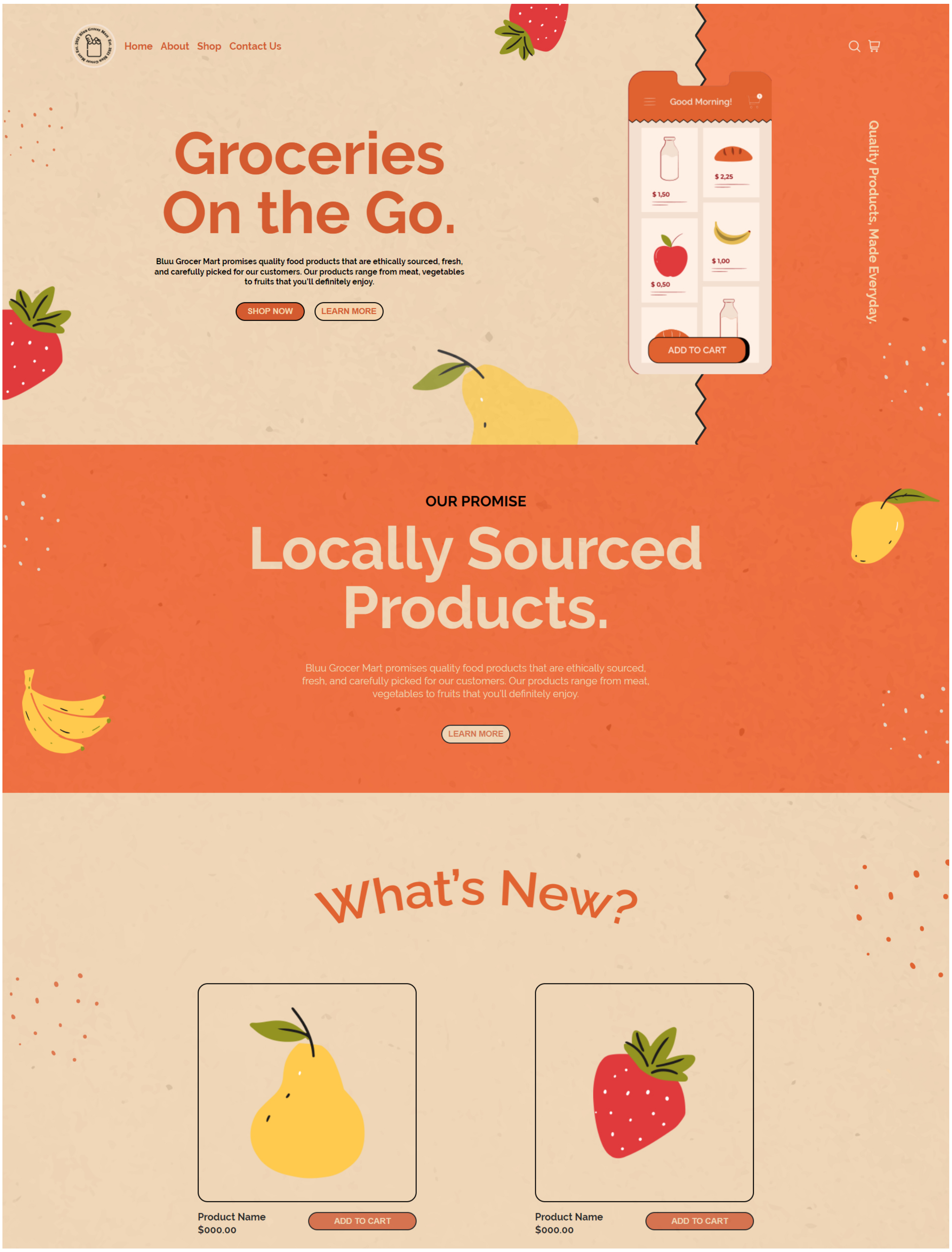
To make the CSS easier to maintain and read, SCSS was used instead of plain CSS, and the stylesheets were split into smaller files and organized in directories. The use of SCSS variables and nesting allowed for cleaner and more organized code.

The resulting web page consists of several sections, including ”Groceries. On the go”, ”Locally sourced block”, ”What’s new”, ”Made simple”, and ”FAQ”. In condition was not specified to implement the ”What’s new” section, but I wanted to try to use the CSS Grid and this looked like a good section to practice it.

During the development process, the main challenge was implementing the background textures and particles while keeping other elements of the design in place. The solution to this problem was to use three layers, with the main class containing the content and texture, and a higher z-index set for the content that needed to be displayed above the texture.

Overall, the laboratory provided a great opportunity to practice and improve coding skills, and learn new techniques for creating and organizing stylesheets.

Annex A: Resulting web page



Groceries On the Go.

Bluu Grocer Mart promises quality food products that are ethically sourced, fresh, and carefully picked for our customers. Our products range from meat, vegetables to fruits that you'll definitely enjoy.

OUR PROMISE

Locally Sourced Products.

Bluu Grocer Mart promises quality food products that are ethically sourced, fresh, and carefully picked for our customers. Our products range from meat, vegetables to fruits that you'll definitely enjoy.

What's New?

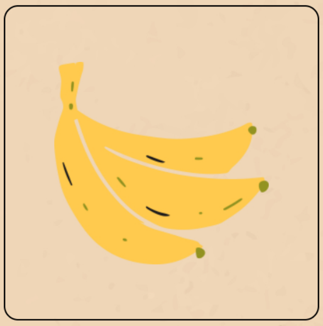
Product Name
$000.00

Product Name
$000.00

Product Name
$000.00
ADD TO CART

Product Name
$000.00
ADD TO CART

Product Name
$000.00
ADD TO CART



Product Name
$000.00
ADD TO CART

Made simple.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Product Name
$000.00
ADD TO CART

Product Name
$000.00
ADD TO CART

Product Name
$000.00
ADD TO CART

← →

FAQs

Do you accept credit card payments? +

What's your refund policy? +

How do deliveries work? +

Do you offer contactless deliveries? +

References

[1] Figpea / PSD extracting tool, *https://figpea.com/* Accessed on February 19, 2023.

[2] Github / valeriaDD / Github repository, *https://github.com/valeriaDD/WebProgramming-lab1* Accessed on February 19, 2023.

[3] Stack Overflow /CSS: Combine Texture and Color, *https://stackoverflow.com/questions/18105833/css-combine-texture-and-color* Accessed on February 19, 2023.