Master DataScience POUR L'ÉCONOMIE ET LA FINANCE

كلية العلوم القانونية والاقتصادية والاجتماعية – طنجة
FSJES TANGER

Final project : E-EPICIER

Implementation of a web app for grocers

Prepared by :

Youssef AMAJOUT
Imane EDDICHE
Zineb KARFA

Supervised by :

Dr. LOTFI ELAACHAK

25/02/2023

Acknowledgement

We would like to express our gratitude to our professor, Mr. LOTFI ELACHAK, for his teaching and guidance throughout the programming module. We have learned from his expertise and rich experience, which have enabled us to understand the key concepts of programming and to successfully complete this project. We are also grateful for the opportunity to work on this project, which has allowed us to apply and develop the skills we have acquired during the module. This has reinforced our understanding of the concepts and has enabled us to develop useful skills for our future careers and to have a versatile profile.

We would also like to thank our families and friends for their constant support and encouragement.

Table of contents

Introduction

In a constantly evolving world, businesses must constantly adapt to technological changes and new trends to remain competitive. The rise of web technologies has opened up new opportunities for businesses in all sectors by enabling them to provide more efficient and convenient services for their customers.

In this context, our project is to develop a full-stack web application based on the Django framework, aimed at grocers. The main objective of this application is to simplify, modernize, and organize the management of customer credits for grocers by providing them with an automated and efficient solution. To achieve this goal, our application must include several daily and essential functionalities such as secure authentication, customer management, product management, and credit management per customer.

To meet the requirements of our specifications, we will use design and development tools such as Figma, Django, Bootstrap, and MySQL. We will also work on creating an intuitive, easy-to-use, and responsive user interface for application users.

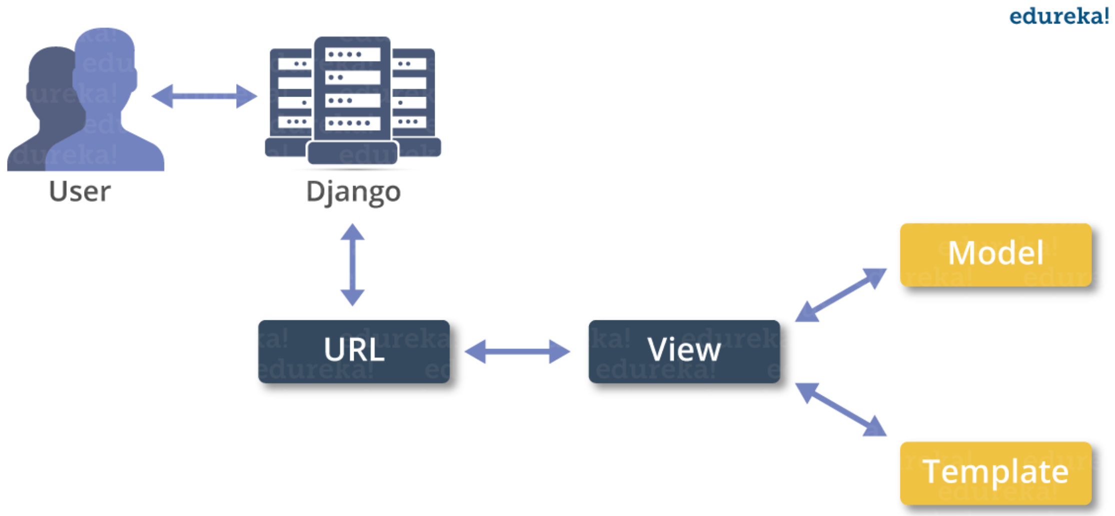
Moreover, working on setting up a robust authentication system to ensure the security and confidentiality of user data is a must. We will use Django's features to create a customer management interface that allows grocers to easily manage the information of all their customers, including their credits.

We will also integrate product and article management to allow grocers to easily manage their stocks and sales. Credit management per customer is an essential feature of the application, which allows grocers to track their customers' credits and manage refunds easily, and differs from their traditional management which creates daily problems for them in managing products or in their relationships with customers.

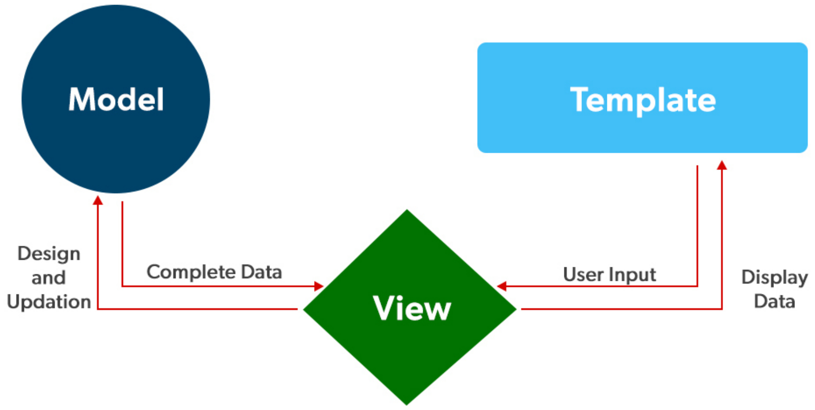
I. Tools used :

During our project, we used a range of tools and technologies to develop our full-stack web application.

- **Python :** is the main programming language used in the development of this project. It is a powerful and versatile programming language that is widely used in web development, data analysis, and scientific computing. In this project, Python was used with Django to build the back-end of the website. It was also used to write the backend code that handles the data processing and management, user authentication and authorization, and communication between the server and the client. Python's rich set of libraries and modules, were also used to perform complex data analysis, visualization, and manipulation tasks. Furthermore, Python's integration with other technologies, such as MySQL and JavaScript, made it easy to incorporate these technologies into the project. Python's popularity and strong community support also provided us with access to a large number of resources, tutorials, and libraries, which makes it easier to develop web applications efficiently. It was used with Django to build the back-end of the website
- **Django** : provides a high-level framework for building web applications in Python. It is a pre-written code library that provides a standard structure and set of functionalities for developing applications. It is a tool that allows developers to write code for specific tasks without having to start from scratch every time. Frameworks provide a set of reusable components, modules, and libraries that help developers to build software applications more efficiently and effectively. Frameworks also provide a standardized way of developing applications, making it easier for developers to collaborate on projects and to maintain code over time. Some popular frameworks include Django, Ruby on Rails, Flask, and Laravel.

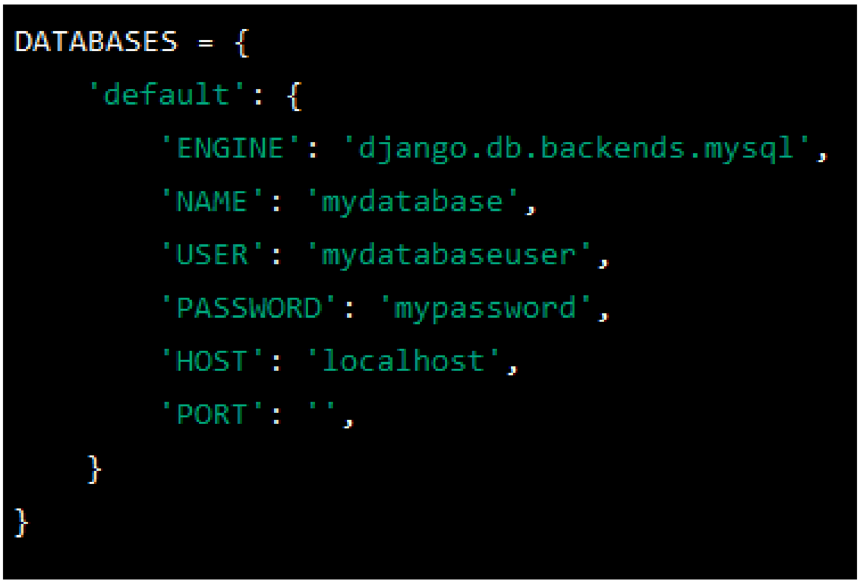


Django follows the model-template-view (MTV) architectural pattern. It provides a set of components and tools that help developers build scalable and secure web applications, including an ORM (Object-Relational Mapping) for database management, an authentication system, a built-in admin interface, and support for URL routing, form handling, and templating. Django is also known for its emphasis on reusability and the principle of "Don't Repeat Yourself" (DRY), which encourages developers to write code that is reusable and maintainable. It is a popular choice for building complex web applications, content management systems, and e-commerce websites.



- **MySQL** : was used as the database management system to store and manage the data of the application. MySQL is an open-source relational database management system that is widely used in web development due to its reliability, scalability, and ease of use.
  To use MySQL, we needed to install and configure it on our server. Then we create a database and tables to store our application data. In the Django framework, we can define the database settings in the `settings.py` file of our project.

Here is an example of how we can define MySQL in a Django project:



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'mydatabase',
        'USER': 'mydatabaseuser',
        'PASSWORD': 'mypassword',
        'HOST': 'localhost',
        'PORT': '',
    }
}
```

The `ENGINE` setting specifies that we are using MySQL as the database backend. The `NAME` setting specifies the name of the database we want to use, while the `USER` and `PASSWORD` settings specify the credentials for accessing the database. The `HOST` setting specifies the location of the database server (in this case, it's on the local machine), and the `PORT` setting specifies the port number to connect to the database server.

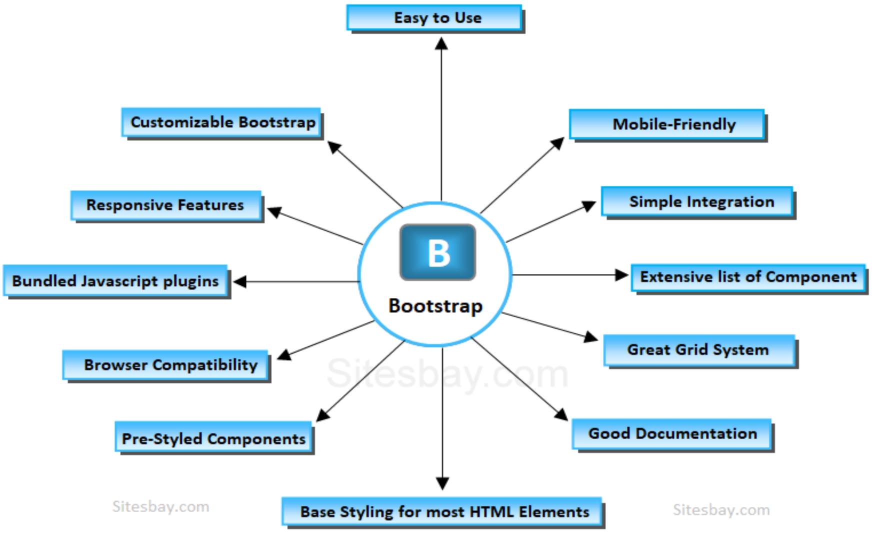
Once we have defined the database settings, we can use Django's ORM (Object-Relational Mapping) to interact with the database and perform CRUD (Create, Read, Update, Delete) operations on our data.

- **Bootstrap** : is a popular open-source framework for building responsive and mobile-first websites and web applications. It was developed by Twitter and is now maintained by the Bootstrap team.

  Bootstrap provides a set of pre-designed HTML, CSS, and JavaScript components, such as buttons, forms, navigation menus, modals, and many others, that can be easily integrated into the project. These components are designed to be mobile-friendly and to adapt to different screen sizes, which means that the website or application will look good and function well on any device, including desktops, tablets, and smartphones.

  In addition to the pre-designed components, Bootstrap also provides a powerful grid system that allows us to create flexible layouts for our pages. The grid system is based on a 12-column layout, which can be customized to meet our needs. We can use the grid system to create different column configurations for different screen sizes, or to create complex, multi-column layouts. It also includes a wide range of customizable CSS and JavaScript utilities that can be used to further style and enhance the website or application. These utilities include responsive typography, spacing, colors, and many others.

Overall, Bootstrap is a powerful tool for web developers who want to create professional-looking, responsive websites and web applications quickly and easily. Its popularity, extensive documentation, and active community make it a great choice for beginners and experienced developers alike.



- **HTML** : is used for creating the structure and content of web pages. It is the markup language that defines the various elements and their placement on the web page, such as text, headings, images, links, and forms. The HTML files in this project serve as the backbone of the website, providing the structure for the content and defining the relationships between various elements. HTML is used in conjunction with CSS to style the web page and create a visually appealing design. Additionally, it is used to create forms that allow users to interact with the website and submit data. These forms may include fields for entering text, checkboxes, radio buttons, and dropdown menus.

- **CSS** : which stands for Cascading Style Sheets, was used extensively in this project to define the presentation and layout of the website's pages. It allowed us to apply styles, such as colors, fonts, spacing, and positioning to HTML elements, making the website more visually appealing and user-friendly. We used it to customize the appearance of the Bootstrap components used in our website, giving the site a unique and consistent style throughout. We also used CSS media queries to make the website responsive to different screen sizes, ensuring that it looks good and functions well on desktop, tablet, and mobile devices.In addition to writing custom CSS code, we also took advantage of the pre-existing CSS rules provided by Bootstrap, which enabled us to create responsive and mobile-first designs with minimal effort. Overall, CSS played a critical role in enhancing the look and feel of our website and ensuring that it is user-friendly and accessible on various devices.

- **Git** : was used for version control. It is a free and open-source distributed version control system that enables developers to track changes to their code.

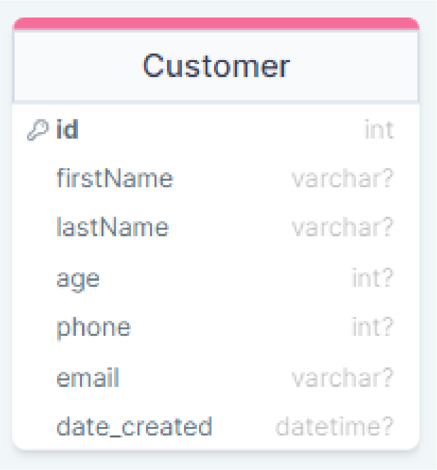
- **GitHub** : was used to manage our code repository. It is a web-based platform used for version control and collaborative software development. It provides a way for developers to store and manage their code, as well as track changes to it over time. In our project, GitHub is used to store the code for the web application developed using Django, as well as any associated files and documentation. This allowed us as a team to easily collaborate on the project, with each member having access to the latest version of the code and the ability to suggest changes or modifications. GitHub can also be used to manage issues and track bugs, allowing the team to stay organized and work towards resolving any issues in a timely manner. Additionally, it provides a way to host the project publicly, allowing other developers to view and potentially contribute to the project. Moreover, using GitHub has provided a streamlined and efficient way for us to work together and manage the project.

- **Visual Studio Code** : The integrated development environment (IDE) used in this project, it is a lightweight and powerful code editor that offers a range of features to enhance the development experience. It is an open-source tool developed by Microsoft and has become one of the most popular code editors among developers. In this project, Visual Studio Code was used to write and edit the code for the application. It provides a wide range of features such as syntax highlighting, auto-completion, debugging, and source control integration. The latter feature is particularly important, as it allowed the team to manage the project's codebase using Git and GitHub. In addition, Visual Studio Code supports a wide range of extensions that can be used to extend its functionality. For example, we have used extensions for Django and Python development, as well as for front-end development with HTML and CSS. It provided a reliable and efficient development environment for the team to work on the project.

II. Database digram

- **<u>Table Customer</u>**

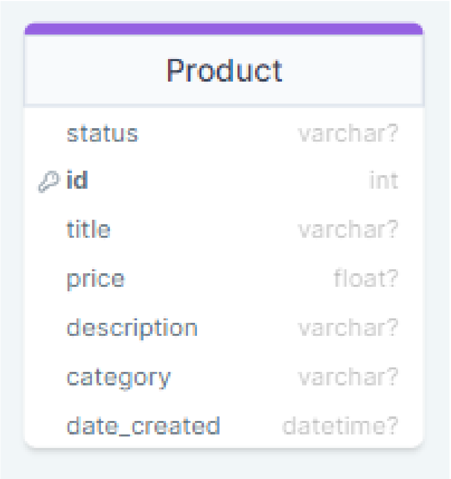
Table Customer is composed of the primary key which is the ID that identifies each customer, along with their details that include first name, last name, age, phone number and email, also the date in which a new customer was inserted in the table.



Customer	
id	int
firstName	varchar?
lastName	varchar?
age	int?
phone	int?
email	varchar?
date_created	datetime?

- **<u>Table Product</u>**

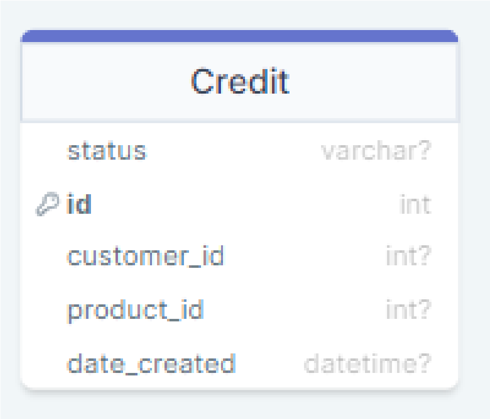
Table Product is composed of the status of the product that defines the choices used in the category, the primary key which is the ID that identifies each product, along with other details including title, price, description, category, also the date in which a new product was inserted in the table.



Product	
status	varchar?
id	int
title	varchar?
price	float?
description	varchar?
category	varchar?
date_created	datetime?

- **Table Credit**

Table Credit is composed of the status of the credit (paid or unpaid), the primary key which is the ID that identifies each credit, along with two foreign keys: the customer ID and the product ID, also the date in which a new credit was inserted in the table.

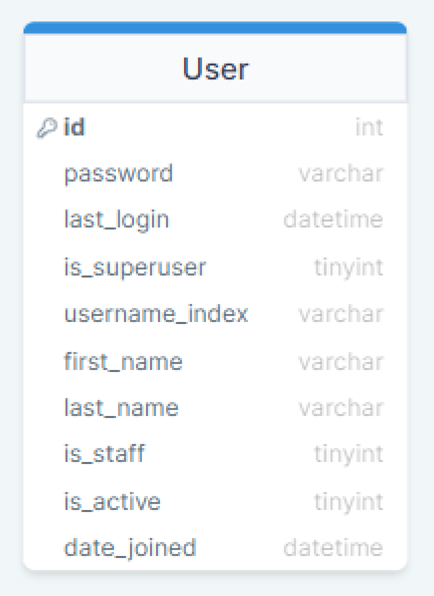


Credit

status	varchar?
id	int
customer_id	int?
product_id	int?
date_created	datetime?

- **Table User**

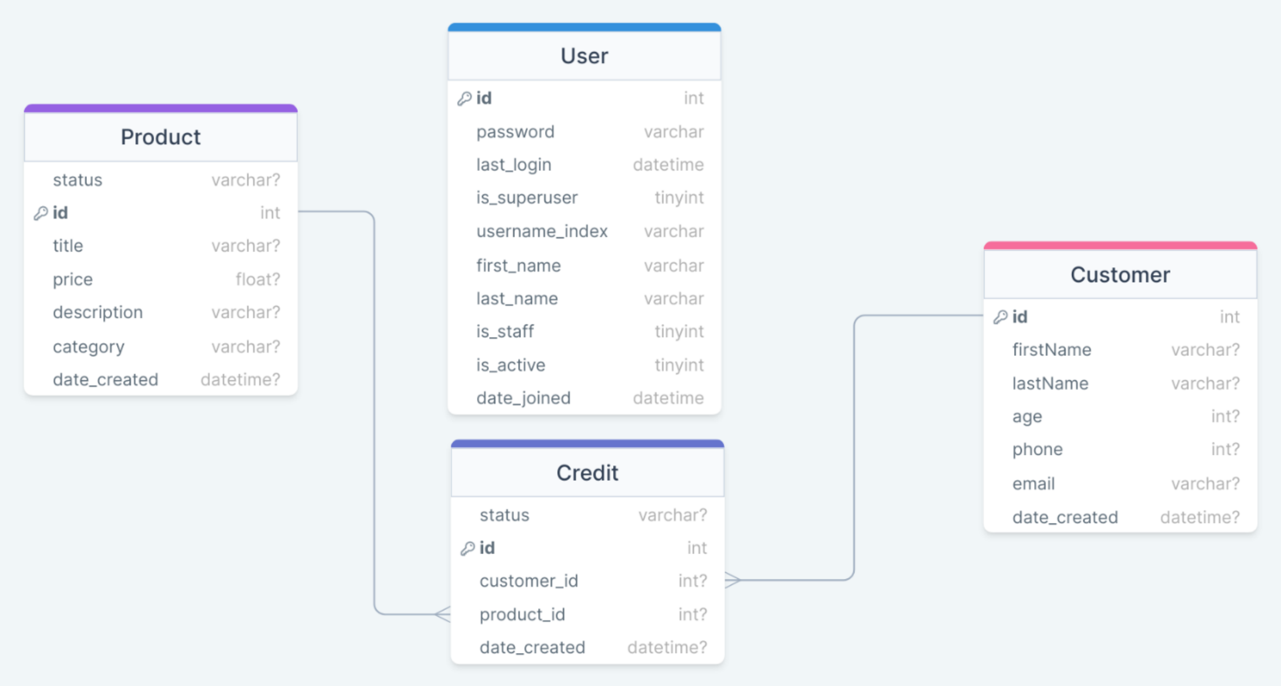
Table User is composed of the the primary key which is the ID that identifies each user, along with the password that is unique for each user, the last login attribute that records the date and time of the user's last login, the “is superuser” boolean attribute indicating whether the user has superuser privileges, the “is staff” boolean attribute indicating whether the user is a member of the staff, the “is active” boolean attribute indicating whether the user account is active, the username, the first name, the last name and the email of the user account, also the date and time when the user account was created.



User	
id	int
password	varchar
last_login	datetime
is_superuser	tinyint
username_index	varchar
first_name	varchar
last_name	varchar
is_staff	tinyint
is_active	tinyint
date_joined	datetime

- **Relations between tables**

The Credit table has two associations with the Customer and Product tables via the Customer ID and the Product ID attributes respectively. These associations are many-to-one relationships as a credit is associated with only one customer and one product, but a customer or a product can have multiple credits. The relationships are represented by two arrows pointing from Customer and Product towards Credit.



Product	
status	varchar?
id	int
title	varchar?
price	float?
description	varchar?
category	varchar?
date_created	datetime?

User	
id	int
password	varchar
last_login	datetime
is_superuser	tinyint
username_index	varchar
first_name	varchar
last_name	varchar
is_staff	tinyint
is_active	tinyint
date_joined	datetime

Credit	
status	varchar?
id	int
customer_id	int?
product_id	int?
date_created	datetime?

Customer	
id	int
firstName	varchar?
lastName	varchar?
age	int?
phone	int?
email	varchar?
date_created	datetime?

III. Development of the web app

Given that Django is installed, you'll need to configure it properly to start building your web application. Here are the basic steps to follow:

1. Create a new Django project : To do so, we open up a terminal and navigate to the directory where we want to store our project. Then, we run the following command:

django-admin startproject projectname

2. Create a new Django app named E-Epicier: Within our project directory, we run the following command to create a new app: **python manage.py startapp appname**

3. Configure the database: Django uses SQLite by default, but we can easily switch to another database like PostgreSQL, MySQL, or Oracle if needed.

To configure Django to use a MySQL database instead of SQLite, we made the following changes in our Django project:

- Install the MySQL Connector for Python: **pip install mysql-connector-python**
- Update the DATABASES setting in our Django project's settings.py file to configure the MySQL database
- Define the tables of the database in the "models.py" file : Customer, Product, Credit
- Run the following command to create the initial schema for our database: **python manage.py makemigrations**
- Run the following command to create the necessary database tables: **python manage.py migrate**

4. Set the static and media roots: The static root is the location of our static files (CSS, images, etc.), while the media root is the location of user-uploaded files. In settings.py, we set the STATIC_ROOT and MEDIA_ROOT variables to the desired locations.

5. Define the URL patterns: To control the flow of our application, we needed to define URL patterns in the urls.py file in our project directory. These patterns map URLs to views in our application.

6. Create views and templates: In Django, views are Python functions that handle HTTP requests. They are defined in the views.py file in our app directory. Templates define the HTML structure of our pages and can be stored in the templates directory within our app.

7. Run migrations: Django uses migrations to manage changes to our database schema. To apply any changes we've made, we run the following command: **python manage.py migrate**

8. Test our application: Finally, we start the development server by running the following command: **python manage.py runserver**

➢ Now, we can access our application by visiting http://localhost:8000/ in any web browser.

IV. How to run the web app :

1- Create a new folder named “epicier” and open Command Prompt in it

2- Download this application using git: https://github.com/Amajout/Demo

3- Create a virtual environment: python -m venv env

4- Activate the virtual environment: .\env\Scripts\activate

5- Install requirements : pip install -r requirements.txt

6- Editing the settings.py file :

```
DATABASES = {
   'default': {
       'ENGINE': 'django.db.backends.mysql',
       'NAME': 'epicier',
       'USER': 'root',
       'PASSWORD':",
       'HOST':'localhost',
   }
}
```

7- Go to MySQL admin and create a database with the same name in the settings file

8- Create migrations: python manage.py makemigrations

9- Migrate migrations : python manage.py migrate

10- Create superuser: python manage.py createsuperuser (choose a username and password)

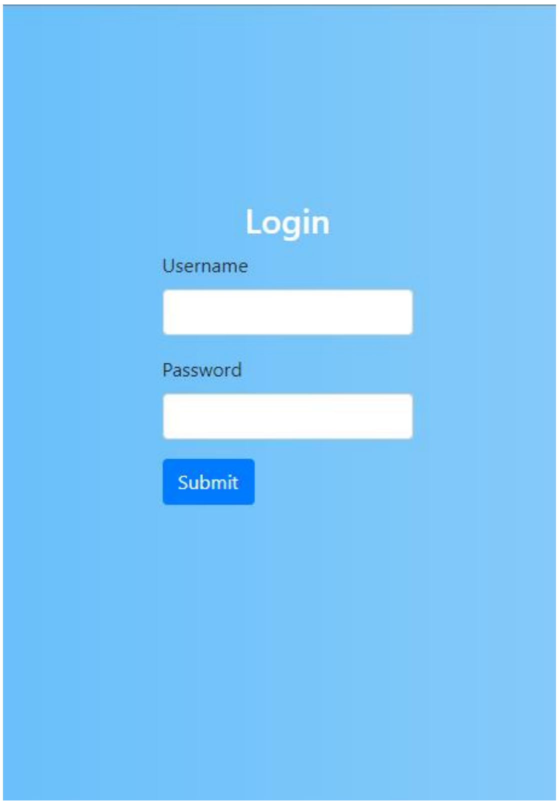
11- Run the project: python manage.py runserver

V. Application interfaces :

1- Login Page :

The login page of our web app serves as a gateway to access the app's features. To log in, you should enter your username and password in the designated input fields.

In the event that a user enters incorrect login information, an error message will be displayed on the page, and they will be prompted to try again

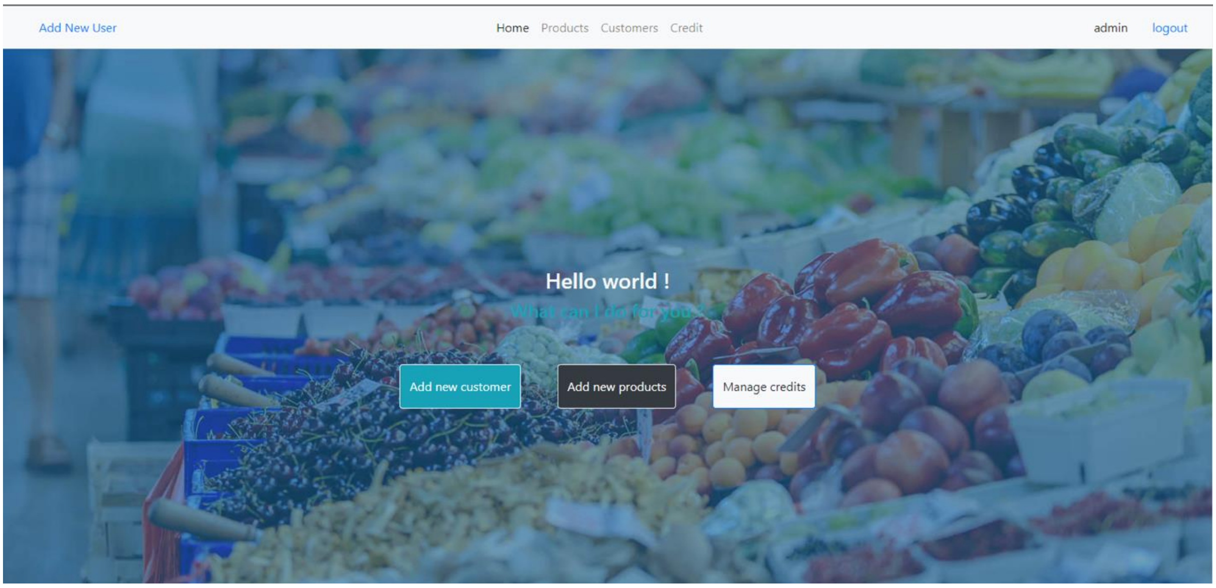


2- Home page :

The app homepage features a header menu with four options: Home, Products, Customers, and Credits. You can navigate to these sections of the app by selecting the corresponding option from the menu. You can also add a new user and logout from the top menu.

In the body of the page, there are three buttons that allow users to perform various actions. The "Add new customer" button enables users to create a new customer profile, while the "Add new product" button allows them to add a new product to the app's inventory. The "Manage credits" button gives users access to a dashboard where they can view and manage customers credits.

Overall, the app homepage provides users with easy access to key sections of the app, as well as convenient buttons for performing common tasks related to managing customers, products, and credits.



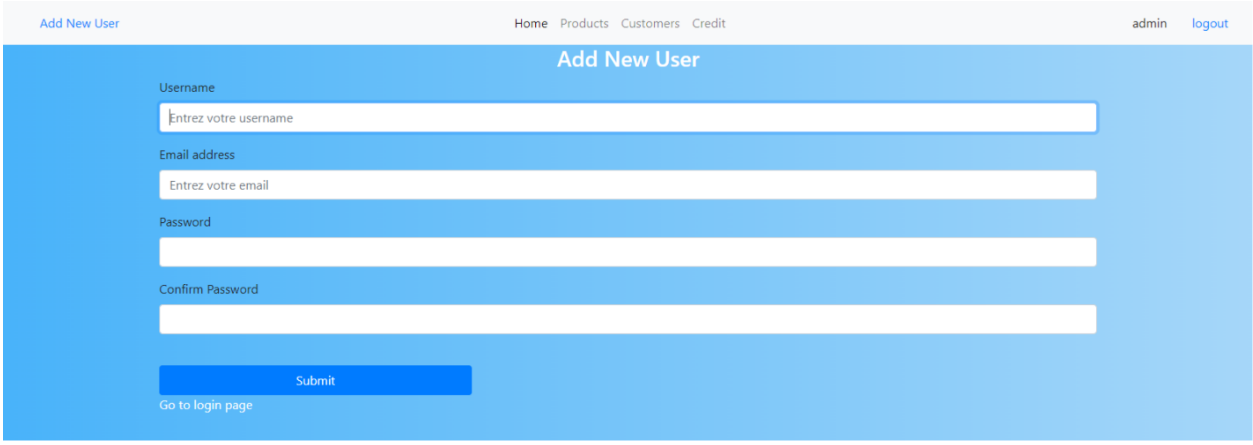
3- Adding a new user :

The add new user page allows administrators to create new user accounts for the app. There are four input fields on the page: one for the username, one for the email address, one for the password, and one for password confirmation.

To create a new user account, administrators must enter a unique username and a valid email address in the corresponding fields. They must also create a password and confirm it by entering it again in the confirmation field.

Once all the required fields have been filled out, administrators can submit the form by clicking on the "Submit" button.

Overall, the add new user page provides administrators with a simple and secure way to create new user accounts for the app.



Add New User

Username

Entrez votre username

Email address

Entrez votre email

Password

Confirm Password

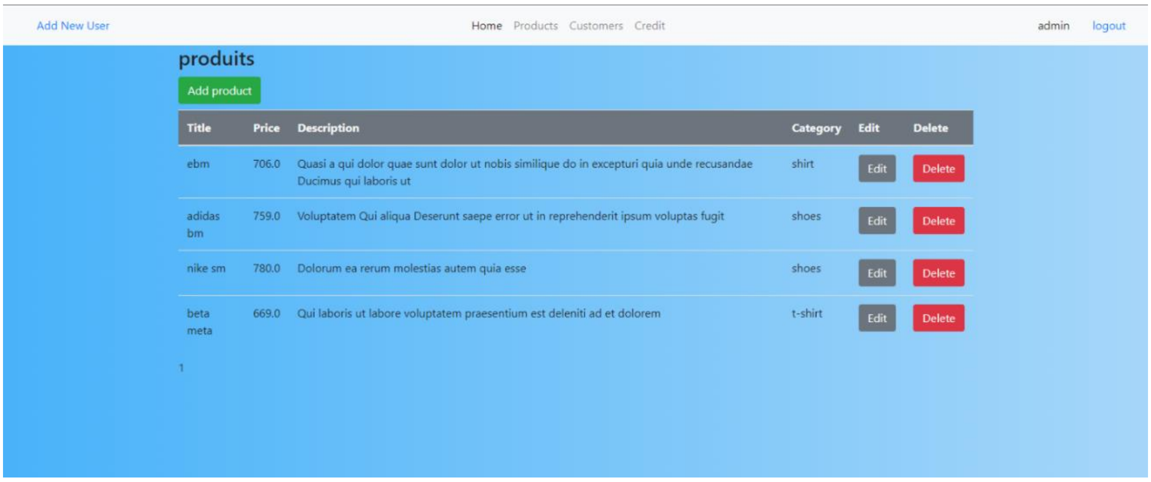
4- Products page :

The products page displays a list of all available products in the app's inventory. The page includes an "Add product" button that allows administrators to add new products to the inventory.

Each product in the list is displayed with its title, price, description, and category, making it easy for users to find and select the product they are looking for. Additionally, each product is accompanied by an "Edit" and a "Delete" button, which allow administrators to modify or remove products from the inventory as needed.

To add a new product, administrators can click on the "Add product" button and fill out the required fields, including the product's title, price, description, and category. Once all the required fields have been completed, the new product can be saved and will appear in the list.

Overall, the products page provides administrators with an easy-to-use interface for managing the app's inventory, with features for adding, editing, and deleting products as needed.



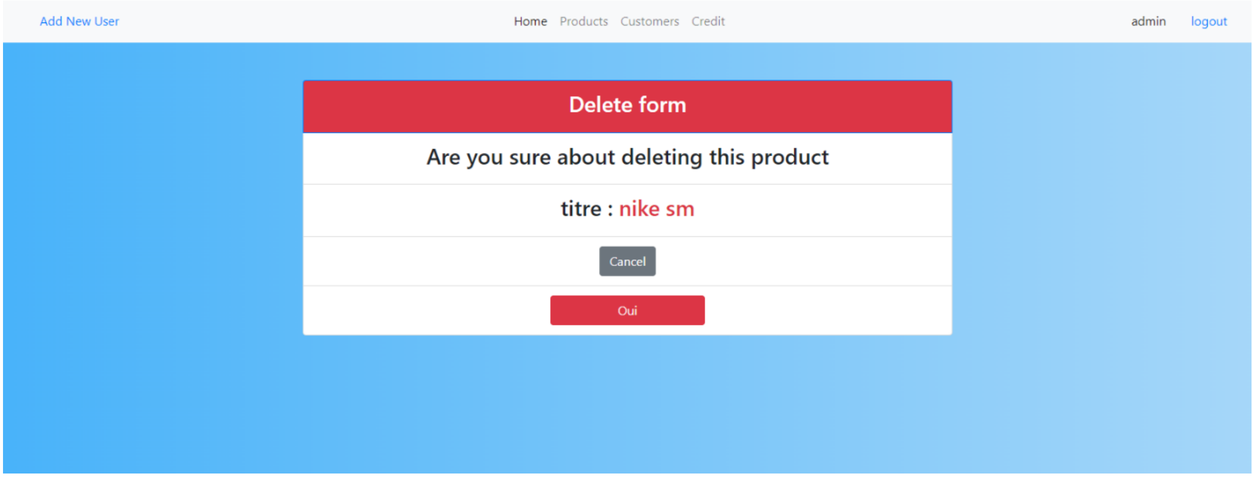
produits

Title	Price	Description	Category	Edit	Delete
ebm	706.0	Quasi a qui dolor quae sunt dolor ut nobis similique do in excepturi quia unde recusandae Ducimus qui laboris ut	shirt	Edit	Delete
adidas bm	759.0	Voluptatem Qui aliqua Deserunt saepe error ut in reprehenderit ipsum voluptas fugit	shoes	Edit	Delete
nike sm	780.0	Dolorum ea rerum molestias autem quia esse	shoes	Edit	Delete
beta meta	669.0	Qui laboris ut labore voluptatem praesentium est deleniti ad et dolorem	t-shirt	Edit	Delete

1

5- Customer Page

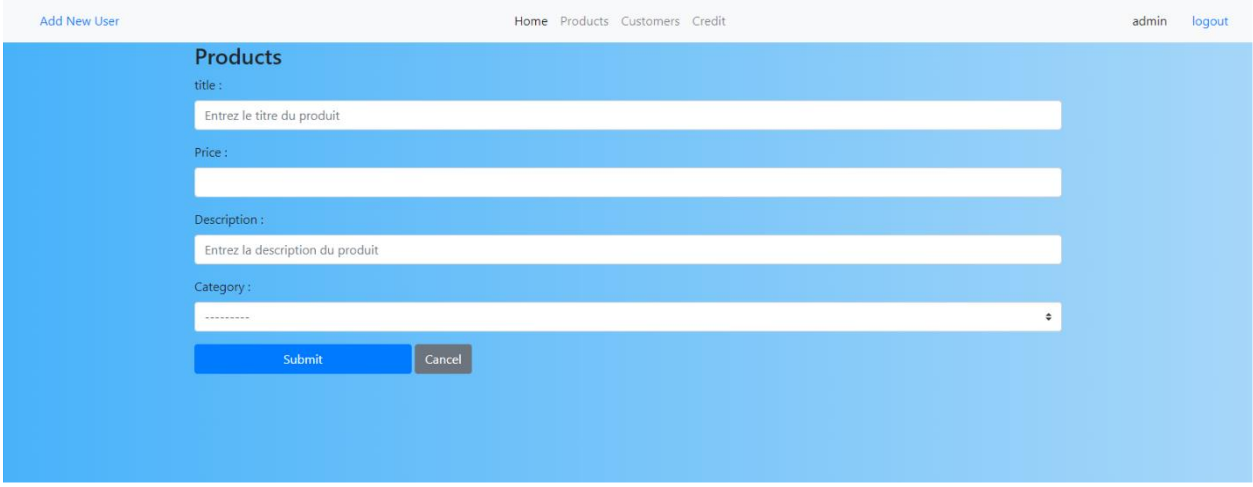
The customer page is where you can manage your customer list. This page displays all of your customers' information, including their first name, last name, age, phone number, and email address. You can also view additional details about each customer by clicking on the "details" button.



Delete form

Are you sure about deleting this product

titre : nike sm



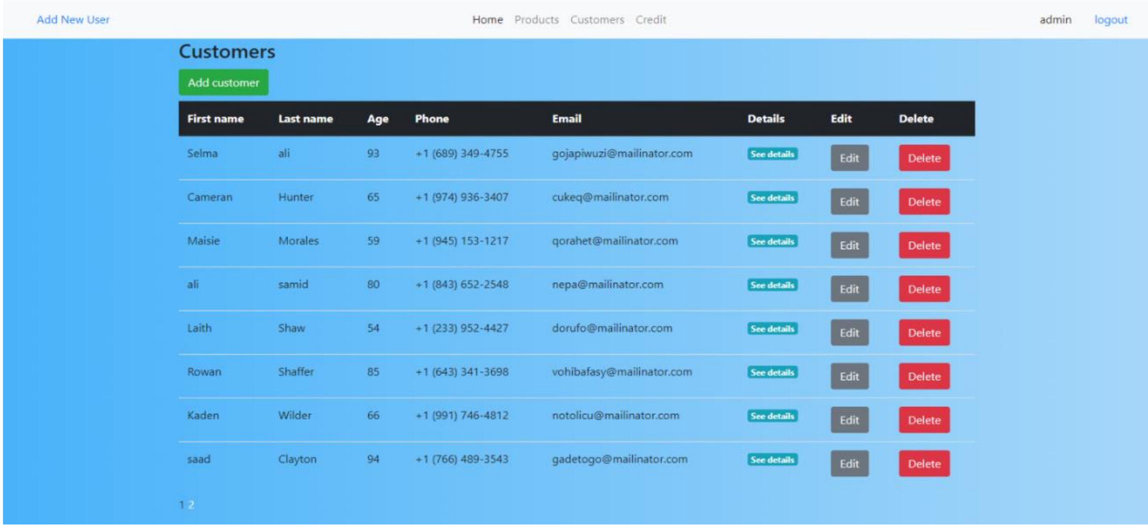
Products

title : Entrez le titre du produit

Price :

Description : Entrez la description du produit

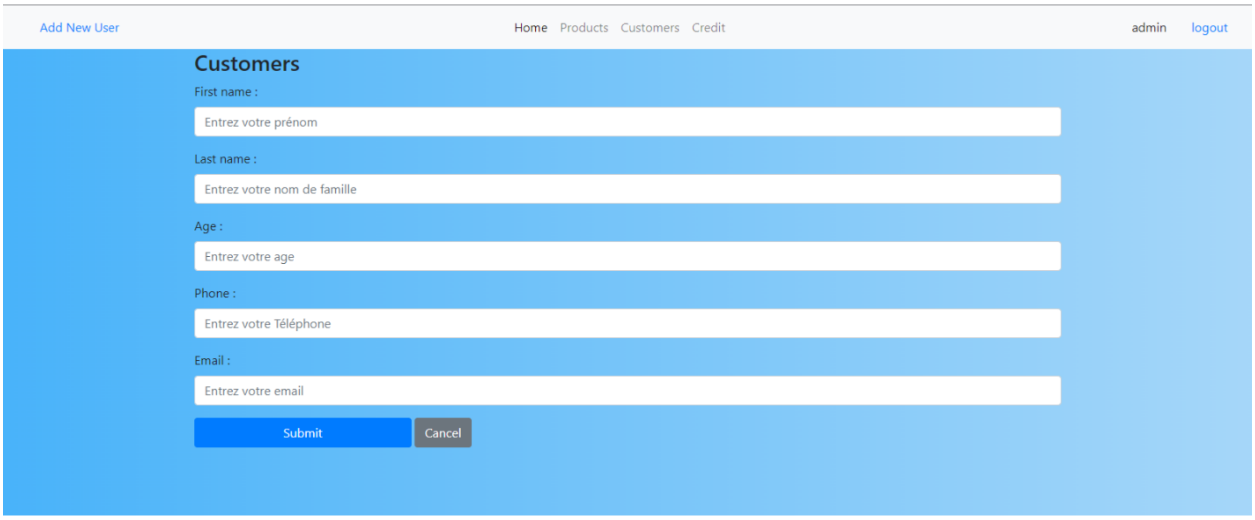
Category : ---------



Customers

First name	Last name	Age	Phone	Email	Details	Edit	Delete
Selma	ali	93	+1 (689) 349-4755	gojapiwuzi@mailinator.com	See details	Edit	Delete
Cameran	Hunter	65	+1 (974) 936-3407	cukeq@mailinator.com	See details	Edit	Delete
Maisie	Morales	59	+1 (945) 153-1217	qorahet@mailinator.com	See details	Edit	Delete
ali	samid	80	+1 (843) 652-2548	nepa@mailinator.com	See details	Edit	Delete
Laith	Shaw	54	+1 (233) 952-4427	dorufo@mailinator.com	See details	Edit	Delete
Rowan	Shaffer	85	+1 (643) 341-3698	vohibafasy@mailinator.com	See details	Edit	Delete
Kaden	Wilder	66	+1 (991) 746-4812	notolicu@mailinator.com	See details	Edit	Delete
saad	Clayton	94	+1 (766) 489-3543	gadetogo@mailinator.com	See details	Edit	Delete

If you need to make changes to a customer's information, you can use the "Edit" button to update their details. If you no longer need a customer's information, you can use the "Delete" button to remove them from your list.



Customers

First name :
Entrez votre prénom

Last name :
Entrez votre nom de famille

Age :
Entrez votre age

Phone :
Entrez votre Téléphone

Email :
Entrez votre email

In addition, you can easily add a new customer to your list by clicking on the "Customer" button, which will prompt you to enter the new customer's details. With this page, you can keep track of all your customers and their details in one convenient location.

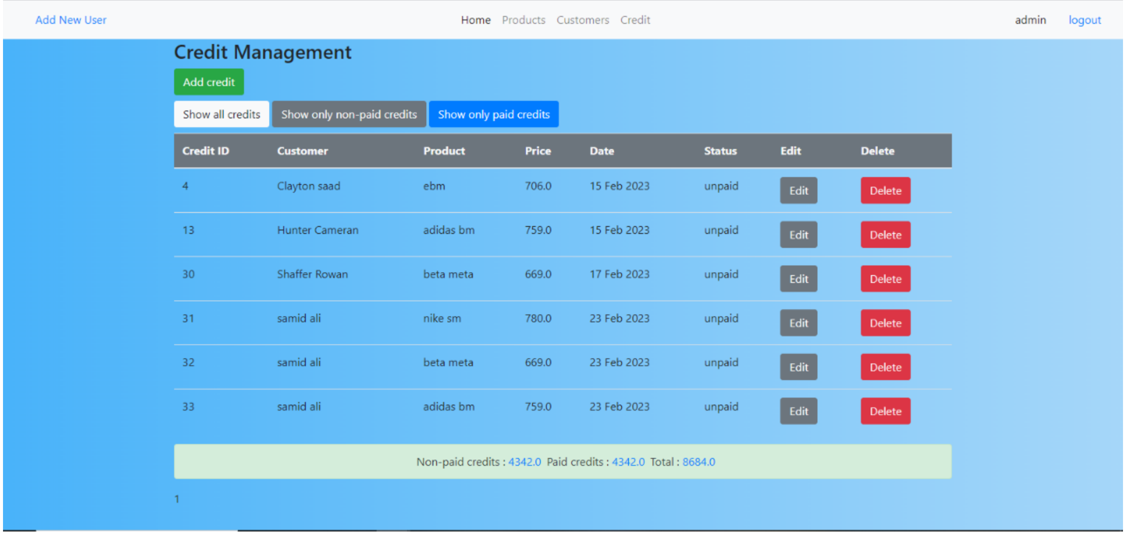
6- Credit Management Page

The credits management page displays a list of all credits in the app's system, along with information about the customer, product, price, date, and payment status. The page also includes an "Add credit" button, as well as "Edit" and "Delete" buttons for each credit in the list.

To help users navigate the list of credits, there are three buttons for filtering the results: "Show all credits", "Show only paid credits", and "Show only non-paid credits". This allows users to easily view credits based on their payment status.

Each credit in the list is identified by a unique credit ID, and is associated with a customer and product name, making it easy to identify which credits correspond to which customers and products. The price and date of each credit are also displayed, along with a payment status indicating whether the credit has been paid or not.

To add a new credit, users can click on the "Add credit" button and fill out the required fields, including the customer name, product, price, and payment status. Once all the required fields have been completed, the new credit can be saved and will appear in the list.



Credit Management

Credit ID	Customer	Product	Price	Date	Status	Edit	Delete
4	Clayton saad	ebm	706.0	15 Feb 2023	unpaid	Edit	Delete
13	Hunter Cameran	adidas bm	759.0	15 Feb 2023	unpaid	Edit	Delete
30	Shaffer Rowan	beta meta	669.0	17 Feb 2023	unpaid	Edit	Delete
31	samid ali	nike sm	780.0	23 Feb 2023	unpaid	Edit	Delete
32	samid ali	beta meta	669.0	23 Feb 2023	unpaid	Edit	Delete
33	samid ali	adidas bm	759.0	23 Feb 2023	unpaid	Edit	Delete

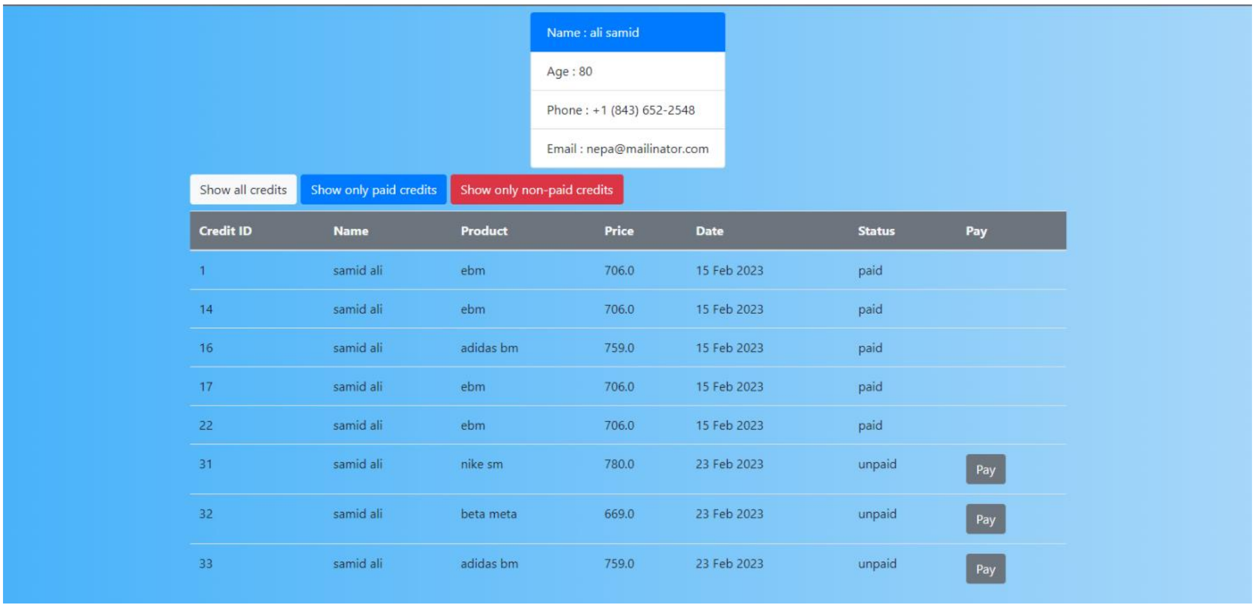
Non-paid credits : 4342.0 Paid credits : 4342.0 Total : 8684.0

1

To modify or delete an existing credit, users can click on the "Edit" or "Delete" button associated with each credit. This allows users to update the details of the credit or remove it from the list entirely.

Finally, the page displays the total amount of credits for both paid and unpaid credits at the bottom of the list.

Overall, the credits management page provides users with a convenient way to view, add, edit, and delete credits within the app, with features for filtering and displaying key information about each credit.



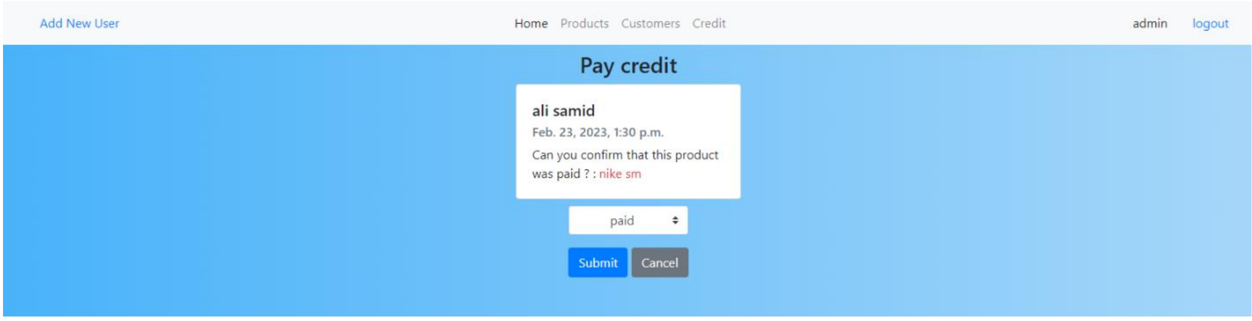
Name : ali samid

Age : 80

Phone : +1 (843) 652-2548

Email : nepa@mailinator.com

Credit ID	Name	Product	Price	Date	Status	Pay
1	samid ali	ebm	706.0	15 Feb 2023	paid	
14	samid ali	ebm	706.0	15 Feb 2023	paid	
16	samid ali	adidas bm	759.0	15 Feb 2023	paid	
17	samid ali	ebm	706.0	15 Feb 2023	paid	
22	samid ali	ebm	706.0	15 Feb 2023	paid	
31	samid ali	nike sm	780.0	23 Feb 2023	unpaid	Pay
32	samid ali	beta meta	669.0	23 Feb 2023	unpaid	Pay
33	samid ali	adidas bm	759.0	23 Feb 2023	unpaid	Pay



Pay credit

ali samid

Feb. 23, 2023, 1:30 p.m.

Can you confirm that this product was paid ? : nike sm

paid

Conclusion

In conclusion, our project to develop a full-stack web application based on the Django framework has been a success. We were able to create an application that meets the needs of grocers by providing them with an effective solution to manage all of their customers' credits and to help them digitize and work in a more modern and organized way.

We worked on implementing a robust authentication system to ensure the security and confidentiality of user data. We also designed a user-friendly interface that allows users to easily navigate in the application and access essential features such as customer management, product or article management, and credit management per customer.

We used different tools such as Figma to design the user interface and Django to develop the application logic. We also integrated the Bootstrap library to create a responsive user interface that is compatible with different types of devices.

Finally, we used the MySQL database to store and manage application data. We worked hard to ensure the quality and reliability of our code and performed rigorous testing to identify and correct potential errors.

In summary, we are pleased with the final result of our project and hope that our application will be useful to grocers in managing their customers' credits and their daily traditional work.

Reference

- Tekki Web Solutions. (n.d.). Bootstrap Web Development: Benefits, Tools, Frameworks, and More. Retrieved from https://www.tekkiwebsolutions.com/bootstrap-web-development/

- "Python Web Development - Django Tutorial." GeeksforGeeks, www.geeksforgeeks.org/python-web-development-django-tutorial/.

- [Edureka] "Django Tutorial - A Comprehensive Guide for Beginners" by Aayushi Johari, last updated on May 4, 2021. Available at: https://www.edureka.co/blog/django-tutorial/